

AML verification

– technical specification of API

1. CHANGE HISTORY	4
2. INTRODUCTION	6
3. VERIFICATION FLOW	6
3.1. AUTHORIZATION	6
3.2. INITIALIZATION	7
3.2.1. CHECKING REGISTRATION	7
3.2.2. INITIALIZING VERIFICATION	8
3.3. SELLER DATA	9
3.4. ASSOCIATES DATA	13
3.4.1. AUTHOLOGIC	15
3.5. UPLOADING DOCUMENTS	16
3.5.1. UPLOADING SCAN OR PICTURE OF DOCUMENT	16
3.5.2. DESCRIBING DOCUMENT	17
3.5.2.1. COMPANY'S DOCUMENTS	17
3.5.2.2. ASSOCIATES' DOCUMENTS	18
3.6. PAYOUTS DETAILS	19
3.6.1. VERIFICATION TRANSFER SERVICE	22
3.6.2. BANK TRANSFER WITHOUT VERIFICATION TRANSFER SERVICE	24
3.6.3. BANK STATEMENT	25
3.6.4. PAYONEER.....	26
3.6.5. WORLDFIRST	26
3.7. NOTIFICATION THAT COLLECTING DATA AND DOCUMENTS HAS BEEN FINISHED	27
3.8. CANCELLATION	27
3.9. GETTING STATUS	28
3.9.1. STATUS CHANGE NOTIFICATION	28
3.9.2. NOTIFICATION RENEWAL	28
3.9.3. OBTAINING STATUS	29
3.9.3.1. POSITIVE	29
3.9.3.2. NEGATIVE	30
3.9.3.3. REJECTED	31
3.9.3.4. WAITING FOR DATA.....	32
3.9.3.5. WAITING FOR VERIFICATION	32
3.9.3.6. CANCELLED	32
3.10. RETRIEVING SELLERS REGISTRY DATA.....	33
3.11. MULTI-ACCOUNTING.....	35
3.11.1. ADDING ADDITIONAL BANK ACCOUNTS FOR SPECIFIC SELLERID	35
3.11.2. ADDING BANK ACCOUNT DETAILS.....	36
3.11.3. COMPLETING VERIFICATION.....	36
3.11.4. REMOVING BANK ACCOUNT	37
3.12. RE-VERIFICATION.....	37
3.13. ID REPLACEMENT PROCESS FLOW	39
3.14. OFFBOARDING	40
3.14.1. OFFBOARDING INITIALIZED BY MARKETPLACE	40
3.14.2. OFFBOARDING INITIALIZED BY PAYU	41
4. VERIFICATION STATUSES AND ERROR CODES	43
4.1. PAYOUT LOCK – NOT IMPLEMENTED YET.....	45
5. EXAMPLES OF REQUESTS	47
6. REFERENCE	51
6.1. GLOBAL PARAMETERS	51
6.2. ENDPOINT SPECIFIC PARAMETERS.....	52

1. Change history

Version	Date	Notes
1.00	2020-01-01	First version
1.01	2020-02-21	VerificationType added in points 3.9.2.1 – 6 Verified added in point 3.6.
1.02	2020-04-17	Status change notification adjusted in 3.9.1
1.03	2020-04-24	<ul style="list-style-type: none"> • Error codes list changed • PEP_STATEMENT document moved to Associates documents • AssociateId - now can be indicated by PayU also • Foreign Account can be defined for payouts
1.04	2020-05-20	<ul style="list-style-type: none"> • bankDataId parameter for loading bank account data added
1.05	2020-07-17	<ul style="list-style-type: none"> • necessary fields length added (seller data and payout data)
1.06	2020-07-31	<ul style="list-style-type: none"> • add more examples of requests
1.07	2020-08-20	<ul style="list-style-type: none"> • Added ErrorCodes 027-031 • Added Verification Transfer service in Payout Details
1.08	2020-10-12	<ul style="list-style-type: none"> • Phone number is no longer required in Seller Data (point 3.3) • Indication in Verification Transfer Service that declared payout data (p. 3.6.1.1) needs to be delivered in 72h since Verification Transfer Service response (p. 3.6.1). • Parameter id change into verificationTransferId in p. 3.6.1
1.09	2020-12-09	<ul style="list-style-type: none"> • Added 032-035 ErrorCodes • Added processType parameter in status (p. 3.9) • Added associateCountryOfBirth parameter to Associates's data • Added descriptions for the Verification Transfer Service • Added length of some fields
1.10	2021-05-28	<ul style="list-style-type: none"> • Added data types • New notification examples
1.11	2021-07-27	<ul style="list-style-type: none"> • Added dateOfBirth for Private persons
1.12	2021-09-03	<ul style="list-style-type: none"> • New attributes added to dataloading/seller • Cancel on every step of verification
1.13	2022-04-21	<ul style="list-style-type: none"> • Added information about sending multipage documents
1.14	2022-05-09	<ul style="list-style-type: none"> • Added PERSONAL_ID_TAX_ID_CHANGE as a verification type (p. 3.2.).
1.15	2022-05-27	<ul style="list-style-type: none"> • Added character limitations of name, email, phone, street, zip, country parameters. • Added Checking Registration section.
1.16	2022-07-27	<ul style="list-style-type: none"> • Added information about PESEL/NIP/IBAN validation. • Added information about error linked to seller not accepting T&C. • Added information about error related to passing invalid IBAN number.
1.17	2022-09-09	<ul style="list-style-type: none"> • Corrected verification-advice endpoints (p. 3.2.1)
1.18	2022-09-19	<ul style="list-style-type: none"> • Added verificationTransferStatus field in the response to the verification status retrieve request.

1.19	2022-12-22	Added new endpoint for adding submerchant with Worldfirst account.
1.20	2023-03-20	<ul style="list-style-type: none"> Added new verification type – “Conversion” Added 036-037 error codes
1.21	2023-05-19	<ul style="list-style-type: none"> Added Parameter Reference in points 6.1, 6.2
1.22	2023-07-14	<ul style="list-style-type: none"> Added Multi-accounting section Added error code 038 – Blacklisted country
1.23	2025-03-13	<ul style="list-style-type: none"> Added new validation for personalIdentificationNumber/taxId. Added information about Authologic integration. Added information about notification renewal. Added enum values for externalVerificationType parameter. Removed “onDocument” from authologic parameters. Added “ID Replacement Process Flow” section. Added endpoint for retrieving sellers registry data. Added offboarding section.
	2025-10-16	<ul style="list-style-type: none"> associateCountryOfBirth is now required
	2025-11-21	<ul style="list-style-type: none"> add ‘039’ – Missing country of birth error code
1.24	2026-01-23	<ul style="list-style-type: none"> Description for offboarding of inactive sellers added

2. Introduction

PayU is obligated by Anti-Money Laundering law to verify customers in that context. This API allows to verify sellers from marketplaces.

We have following verification types:

- Full AML verification (data of seller + scans or pictures of documents + bank account for withdrawals)
- Only bank account for withdrawals
- Only AML verification (data of seller + scans or pictures of documents)
- Periodical re-verification – full or partial (only indicated data or documents need to be provided)

Note: When documents and data requested by PayU are send (for all types of verification), to start verification process it must be followed by /complete request.

General flow of verification contains following steps:

- Initialization of verification – can be done by both marketplace or PayU, depending on context.
- Marketplace provides seller data and documents.
- Marketplace informs PayU that all verification data have been collected (complete).
- PayU verifies provided data and determines verification status.
- Status of verification is sent to Marketplace. It may inform marketplace about the missing verification data or final verification status.
- If some data is missing, Marketplace must send them to PayU and again inform PayU all verification data have been collected (complete).
- If status is final – verification is closed.

Verification can be initialized by both marketplace or PayU. Most common case for initialization by marketplace is registering of seller unknown to PayU.

Most common case for initialization by PayU is requirement of re-verification - when some of documents have expired or the verification is no longer valid .

3. Verification flow

3.1. Authorization

Every API request must be authorized. Authorization process is described under the link (only Oauth method is allowed):

http://developers.payu.com/en/restapi.html#references_api_signature

During the verification process, please **do not** send multiple requests at the same time. Sending repeated requests without waiting for a response will result in a blocked verification and an error message from PayU.



POS which will be used for authorization should be indicated by Marketplace and should not change.

3.2. Initialization

3.2.1. Checking Registration

Note: This feature requires additional setting on the PayU side.

Endpoint `/verification-advice` allows verifying, on the basis of a Personal Identification Number or NIP number, whether a given individual/company has already been registered in the PayU system.

GET `/api/aml-verification/v1/verification-advice`

This endpoint is designed to verify the registration status of legal entities. To specify whether you want to check status of a company or an individual you have to add a **companyTaxId** or **personalIdentificationNumber** query parameter:

GET `/api/aml-verification/v1/verification-advice?companyTaxId=<tax_id>`

- **companyTaxId** `<string>` - company tax id of entity.

or individuals:

GET `/api/aml-verification/v1/verification-advice?personalIdentificationNumber=<personal_id>`

- **personalIdentificationNumber** `<string>` - personal identification number of individual.

Responses:

- Http 200 – Advised verify action. Possible values:
 - **FULL_ONBOARDING_REQUIRED** - full boarding process required,
 - **BANK_ACCOUNT_VERIFICATION_REQUIRED** - only a verification transfer is required for verification purposes.
- Http 400 – Invalid input query values.
- Http 404 – Marketplace not found.

In case you received **BANK_ACCOUNT_VERIFICATION_REQUIRED** in response, you have to follow some additional steps:

1. Create verification with "`type": "FULL"` – [3.2.2. Initializing Verification](#)
2. Send basic seller data – [3.3 Seller Data](#)
3. Send account data – [3.6.1. Verification transfer service](#)

4. Send verification complete – 3.7. [Notification that collecting data and documents has been finished](#)

After these steps you must make a transfer to trigger verification.

3.2.2. Initializing Verification

You have 90 days from the start of the verification process to submit all required documents. Otherwise, the process will be terminated with a NEGATIVE status.

In order to initialize verification, the following request must be send on `/verification` endpoint:

POST `/api/aml-verification/v1/verification`

```
{  
  "sellerId": "seller_27",  
  "type": "FULL"  
}
```

sellerId `<string>`

- unique id of seller (submerchant) given by marketplace side. This id must be constant for every verification concerning the same seller (submerchant).

type `<string>` - Verification type:

1. PAYOUT_ACCOUNT_DATA - initialized by marketplace side. The verification with this type should contain the data of payout account and seller (submerchant). If this verification initializes seller in PayU then seller data are required. If it is just account update, then only account data are required.
2. FULL - verification which requires providing by seller all data for AML4 and payouts, typically initialized by marketplace side.
3. UPDATE - Verification is typically initialized by the merchant side when the merchant updates his data (e.g. company address). If the merchant changes critical data, e.g. tax id/personal id, legal type, it is required to use PERSONAL_ID_TAX_CHANGE or CONVERSION respectively.
4. REVERIFICATION - verification typically initialized by PayU, when we require merchant to update his data (e.g. no longer valid document).
5. PERSONAL_ID_TAX_ID_CHANGE - initialized by marketplace side. This allows to change PERSONAL_ID for private person or TAX_IDs for legal entity. Once the KYC verification process begins, the current account and payouts are blocked until verification is completed. If the process is interrupted (e.g. the company is unable to verify itself), PayU restores the old merchant account, thus unlocking the payouts on it.
6. CONVERSION - initialized by marketplace side. This verification type allows to change a private person into the legal entity. Once the KYC verification process begins, the current account and payouts are blocked until verification is completed. If the process is interrupted, PayU restores the old seller's account, thus unlocking the payouts on it.

Response for this request will be as follows:

- Http 200 - Success

```
{
  "verificationId": "aZ16Ja",
  "status": "WAITING_FOR_DATA",
  "sellerId": "seller_27"
}
```

- Http 400 – Invalid input data
- Http 409 – Verification may be already in progress

verificationId <long> – unique id of the verification provided by PayU

status <string> – current status of the verification:

- **WAITING_FOR_DATA** - status after initialization of the verification, when PayU is waiting for data or after exceeding transaction limits for the occasional transaction flow
- **WAITING_FOR_VERIFICATION** – set when all the data were send to PayU (after complete)
- **REJECTED** – set when there are some errors in the provided data. These status is send with error codes
- **POSITIVE** – final status for approved verification
- **NEGATIVE** – final rejection status

Obtained verificationId needs to be used for whole communication according this case.

3.3. Seller data

This is method that allows to send data of the seller (submerchant).

If you provide the values for address.country and personalIdentificationNumber/taxId, you can only update them by submitting a /verification request of type PERSONAL_ID_TAX_ID_CHANGE.

POST [/api/aml-verification/v1/dataloading/seller](#)

JSON for company data:

```
{
  "verificationId": "aZx65uY",
  "sellerId": "seller_1",
  "email": "company@test.com",
  "phone": "111222333",
  "taxId": "1121222245",
  "companyName": "Company 1",
  "legalForm": "CIVIL_LAW_PARTNERSHIP",
  "registryNumber": ""}
```

```

"registrationDate": "2019-07-07",
"address": {
    "street": "Grunwaldzka 186",
    "zipcode": "60-166",
    "city": "Poznan",
    "country": "PL"
},
"isAccountCloned": "false"
}

```

JSON for private person:

```

{
    "verificationId": "BZx65uYW",
    "sellerId": "seller_2",
    "email": "test@test.com",
    "phone": "111222333",
    "taxId": "",
    "name": "Jan",
    "surname": "Kowalski",
    "legalForm": "PRIVATE_PERSON",
    "personalIdentificationNumber": "90112322330",
    "dateOfBirth": "1988-10-12",
    "registryNumber": "",
    "address": {
        "street": "Grunwaldzka 186",
        "zipcode": "60-166",
        "city": "Poznan",
        "country": "PL"
    },
    "isAccountCloned": "false"
}

```

Request parameters:

verificationId `<long>` – id of verification in context of which data are provided. Always required

sellerId `<string>` – id of seller (submerchant). Required.

companyName `<string>` – name of the company. Required if seller type is not private person.

name `<string>` – name of seller. Required if seller type is private person. [1, 255 characters]

surname `<string>` – surname of seller. Required if seller type is private person.

taxId `<string>` [string with no spaces eg. 1122334455] – Tax id. Required. Set null for “PRIVATE_PERSON” legalForm. [1, 60 characters]

legalForm `<string>` – (**FOREIGN_COMPANY** type has been deprecated) available values:

- “AGENCY_OF_FOREIGN_COMPANY”,
- “ASSOCIATION”,
- “CHURCH_LEGAL_PERSON”,
- “CIVIL_LAW_PARTNERSHIP”,
- “COOPERATIVE”,

"EDUCATIONAL_INSTITUTION",
 "FOUNDATION", "GENERAL_PARTNERSHIP",
 "JOINT_STOCK_COMPANY",
 "LIMITED_JOINT_STOCK_PARTNERSHIP",
 "LIMITED LIABILITY_PARTNERSHIP",
 "LIMITED_PARTNERSHIP",
 "LIMITED LIABILITY_COMPANY",
 "LOCAL_GOVERNMENT_UNIT",
 "PRIVATE_PERSON",
 "PROFESSIONAL_PARTNERSHIP",
 "LOCAL_GOVERNMENT_CULTURE_INSTITUTION",
 "RESEARCH_INSTITUTE",
 "SOLE_TRADER",
 "STATE_COMPANY",
 "OTHER".

gusCode `<string>` - <https://stat.gov.pl/metainformacje/slownik-pojec/pojecia-stosowane-w-statystyce-publicznej/97/pojecie.html>

registryNumber `<string>` – Company registry number (e.g. REGON in Poland).

registrationDate `<date>` - YYYY-MM-DD company registration date in the registries

address – address of the company or person (for private persons):

street Street with building and apartment number

`<string>` [1, 255 characters]

Zipcode [1, 20 characters]

`<string>`

city [1, 40 characters]

`<string>`

country string (two letter ISO country code)

`<string>` [1, 80 characters]

isAccountCloned `<boolean as string>` - information if the account was cloned for the same seller from the already existing and verified one. Used when marketplace allows the account cloning operation.

email `<string>` - Contact email. Required. [5, 80 characters]

phone `<string>` - Contact phone. [1, 40 characters]

personalIdentificationNumber `<string>` – national, personal identification number e.g. PESEL in Poland. Required for private persons for full verification (**PESEL is required whenever it has been issued**).

dateOfBirth `<date>` - Required for private persons when sending basic data

Responses:

http 204 - Data accepted for further verification.

http 400 – Invalid input. Reason specified in content.

Examples:

Too short **personalIdentificationNumber/taxId**, example for value: 123

Error message:

```
{
  "message": "Too short value: taxId.",
  "code": 400
}
```

Error message:

```
{
  "message": "Too short value: personalIdentificationNumber.",
  "code": 400
}
```

Blacklisted **personalIdentificationNumber/taxId**.

Error message:

```
{
  "message": "Prohibited identifier detected: personalIdentificationNumber.",
  "code": 400
}
```

Error message:

```
{
  "message": "Prohibited identifier detected: taxId.",
  "code": 400
}
```

Incorrect **personalIdentificationNumber/taxId**, example for value: 111111111

Error message:

```
{
  "message": "Invalid input value: taxId.",
  "code": 400
}
```

Error message:

```
{
  "message": "Invalid input value: personalIdentificationNumber.",
  "code": 400
}
```

http 404 – Verification not found.

http 409 – Verification state does not allow operation.

We are checking if seller/iban is on our black list, we are doing it synchronously while receiving the request. In such cases PayU will respond depending on the data sent:

- PESEL: 'Blacklisted Personal ID Number'.
- NIP: 'Blacklisted TAX ID'.

- IBAN: 'Blacklisted Bank Account Number'.

Field	Company	Private person
verificationId	R	R
sellerId	R	R
companyName	R	NA
name	NA	R
surname	NA	R
taxId	R	NA
legalForm	R	R
gusCode	O	NA
registryNumber	O	NA
registrationDate	O	NA
address	R	R
street	O/R	O/R
Zipcode	O/R	O/R
city	O/R	O/R
country	R	R
isAccountCloned	O	O
email	R	R
phone	O/R	O/R
personalIdentificationNumber	NA	R
dateOfBirth	NA	R

R - required

O - optional

O/R - required for full verification

NA - not applicable

3.4. Associates data

On this endpoint data of associates - representatives and beneficiaries according to appropriate instructions from PayU, which is not part of this specification, should be provided.

Only one associate can be added for Private Person legal form.

POST </api/aml-verification/v1/dataloading/associates>

```
{
  "verificationId": "aZx65uY",
  "associateId": "dsada65",
  "associateType": "BENEFICIARY",
  "associateName": "Jan",
  "associateSurname": "Kowalski",
  "associateBirthDate": "2019-01-27",
  "associateIdentityNumber": "90120789112",
  "associateCitizenship": "PL",
```

```

    "associateCountryOfBirth": "PL"
}

```

associateId `<string>` - Associate id. Unique on seller level.

associateType `<string>` - Type of the associate. [BENEFICIARY, REPRESENTATIVE]

associateName `<string>`

associateSurname `<string>`

associateCitizenship `<string>` – two letter ISO country code

associateIdentityNumber `<string>` - Unique associate identification number. PESEL in Poland

PESEL is required whenever it has been issued.

associateBirthDate `<date>` – YYYY-MM-DD

associateCountryOfBirth `<string>` – two letter country code

Field	
verificationId	R
associateId	R
associateType	R
associateName	R
associateSurname	R
associateCitizenship	R
associateIdentityNumber	R
associateBirthDate	O/R
associateCountryOfBirth	R

R - required

O - optional

O/R – optional/required

NA - not applicable

Field **associateIdentityNumber** must contain personal identification number - PESEL in Poland (**PESEL is required whenever it has been issued**). If given person do not possess PESEL then fields **associateCitizenship** and **associateBirthDate** should be provided.

Responses:

http 204 - Success.

http 400 – Invalid input. Reason specified in content.

http 400 – Only one associate can be added for Private person.

http 404 – Verification not found.

http 409 – Verification state does not allow operation.

We are checking if associate is on our black list, we are doing it synchronously while receiving the request. In such cases PayU will respond with:

- PESEL: 'Blacklisted Personal ID Number'.

3.4.1. Authologic

The Authologic service requires additional configuration from PayU, alongside specific requirements that the marketplace platform must meet. Prior to integration, please reach out to PayU to confirm compatibility of your platform with the Authologic service.

If you verified the associates through Authologic, please send additional `externalVerificationData` object containing details about verified documents and `authologicConversationID`.

POST `/api/aml-verification/v1/dataloading/associates`

```
{
  "verificationId" : "{{VERIFICATION_ID}}",
  "associateId" : "TestRepresentative2",
  "associateType" : "BENEFICIARY",
  "associateName" : "Adam",
  "associateSurname" : "Kowalski",
  "associateCitizenship" : "PL",
  "associateIdentityNumber" : "94091100007",
  "associateCountryOfBirth" : "Poland",
  "associateBirthDate" : "2000-09-11",
  "verified": "true",
  "externalVerificationData" : {
    "externalVerificationType" : "MOJEID_PL",
    "documentNumber" : "ABC123456",
    "documentType" : "ID_CARD",
    "documentExpiryDate" : "2030-09-11",
    "identityNumber" : "94091100007",
    "firstName": "Jan",
    "surname": "Nowak",
    "firstNameSecondDocument": "Jan",
    "surnameSecondDocument": "Nowak",
    "nationality" : "PL",
    "dateOfBirth" : "2000-09-11",
    "secondDocumentNumber" : "DFE123456",
    "secondDocumentType" : "PASSPORT",
    "secondDocumentExpiryDate" : "2030-09-11",
    "identityNumberSecondDocument" : "94091100007",
    "nationalitySecondDocument" : "PL",
    "dateOfBirthSecondDocument" : "2000-09-11",
    "authologicConversationID" : <String>
  }
}
```

externalVerificationType – the type of verification that was used in Authologic. Possible values:

- MOJEID_PL,
- MOJEID_CZ,
- BANKID_CZ,
- EDO,
- MOBYWATEL,

- BIOMETRIC_VERIFICATION_OCR.

documentNumber – document number.

documentType – document type.

documentExpiryDate – expiry date of the document.

identityNumber – the identification number of the associate as stated in the document.

nationality – the nationality of the associate as stated in the document.

dateOfBirth – the date of birth of the associate as stated in the document.

secondDocumentNumber – second document number.

secondDocumentType – second document type.

secondDocumentExpiryDate – expiry date of the second document.

identityNumberSecondDocument – the identification number of the associate as stated in the second document.

nationalitySecondDocument – the nationality of the associate as stated in the second document.

dateOfBirthSecondDocument – the date of birth of the associate as stated in the document.

authologicConversationID – external identifier of the Authologic verification process.

3.5. Uploading documents

3.5.1. Uploading scan or picture of document

This method allows to upload these filetypes: PNG, JPEG, JPEG 2000, PDF, BMP, TIFF which maximum file size is 10 MB. There must be two requests for uploading document used: payu_widget or dataloading/files and dataloading/..../documents. First request uploads the file into server whereas second connects the file to the appropriate submerchant, associate and verification. ID documents (ID Cards, Passports, Driving licenses) with multiple pages should be sent as follows:

- front side should be the first element in table ‘files’,
- the back side should be the second element in table ‘files’.

POST [/api/aml-verification/v1/dataloading/files](#)

Multipart parameters:

verificationId

filename

content

Responses:

http 200 – Success

```
{
  "verificationId": "aZx65uY",
  "fileId": "string"
}
```

fileId – identificatory of uploaded file given by PayU. Please save it for whole process of verification – this will be used in dataloading/..../documents request.

http 400 – Invalid input. Reason specified in content.
http 404 – Verification not found.
http 409 – Verification state does not allow operation.

3.5.2. Describing document

3.5.2.1. Company's documents

This method allows to load company related documents.

```
POST /api/aml-verification/v1/dataloading/seller/documents
{
  "verificationId": "<string>",
  "documentId": "<string>",
  "type": "<string>",
  "files": [
    "<string>",
    "<string>"
  ],
  "translationFiles": [
    "<string>",
    "<string>"
  ],
  "verified": "true"
}
```

Parameters:

type `<string>` - Document types:

- REGISTRY_DOCUMENT
- BANK_ACCOUNT AGREEMENT
- INVOICE
- UBO_STATEMENT
- CIVIL_LAW AGREEMENT
- PROXY_DOCUMENT
- OTHER_DOCUMENT

files `<string>` – list of file ids given in a response to the dataloading/files request. There can be many IDs in this field when one document contains many files.

translationFiles `<string>` – fileId of translation of documents (Polish or English documents do not need to be translated)

verified – indicates that the documents for this associate have been verified externally, outside of PayU. The document verification process on the PayU side will be skipped.

Responses:

http 204 - Success.
http 400 – Invalid input. Reason specified in content.
http 404 – Verification not found.
http 409 – Verification state does not allow operation.

3.5.2.2. Associates' documents

For uploading associates – representatives or beneficiaries, documents use the following request:

POST [/api/aml-verification/v1/dataloading/associates/documents](#)

```
{  
  "verificationId": "<long>",  
  "associateId": "<integer>",  
  "type": "<string>",  
  "files": [  
    "<string>",  
    "<string>"  
>,  
  "documentId": "<string>",  
  "documentNumber": "<string>",  
  "issueDate": "<date>",  
  "expireDate": "<date>",  
  "translationFiles": [  
    "<string>",  
    "<string>"  
>,  
  "verified": "true"  
}
```

associateId `<integer>` – unique id of an associate assigned by Marketplace side

documentId `<string>` – unique document ID assigned by Marketplace side

type `<string>` – Document type - [ID_CARD, PASSPORT, DRIVING_LICENCE, RESIDENCE_PERMIT, OTHER_DOCUMENT, PEP_STATEMENT]

documentNumber `<string>` – serial number of provided document

issueDate `<date>` – issue date of document [YYYY-MM-DD]

expireDate `<date>` – expiry date of document [YYYY-MM-DD]

files – list of file ids given in a response to the dataloading/files request. Can be many IDs in this field when one document contains many files.



translationFiles – fileId of translation of documents (Polish or English documents do not need to be translated)

verified – indicates that the documents for this associate have been verified externally, outside of PayU. The document verification process on the PayU side will be skipped.

Field	
verificationId	R
associateId	R
type	R
files	R
documentId	R
documentNumber	O
issueDate	O
expireDate	R
translationFiles	O
verified	O

R - required

O - optional

O/R – optional/required

NA - not applicable

Responses:

http 204 - Success.

http 400 – Invalid input. Reason specified in content.

http 404 – Verification not found.

http 409 – Verification state does not allow operation.

3.6. Payouts details

To make a withdrawal, a bank account number must be set for the submerchant. The submerchant must provide this bank account number, which is then verified by cross-checking the declarative data against the information received from the bank.

Several verification methods are available, and the methods applicable to a particular implementation must be agreed upon with PayU. We recommend using the [Verification Transfer Service](#), as other methods require additional configuration.

The data must be sent to PayU using the appropriate request:

POST [/api/aml-verification/v1/dataloading/payouts/bankAccountData](#)

```
{  
  "bankDataId": "<string>",  
  "verificationId": "<long>",  
  "associateId": "<long>"}  
}
```

```

"accountNumberRequested": "<string>",
"hasDocument": "<boolean as string>",
"swiftCode": "<string>",
"payoutDataVerificationType": "<string>",
"bankName": "<string>",
"bankAddress": "<string>",
"bankCountry": "<string>",
"statementData": {
    "accountNumberFromBank": "<string>",
    "ownerName": "<string>",
    "address": {
        "street": "<string>",
        "zipcode": "<string>",
        "city": "<string>",
        "country": "<string>"
    }
},
"foreign" : "<boolean as string>",
"paymentId" : "<string>",
"verified": "<boolean as string>",
"verificationTransferId": "<string>",
}
}

```

List of parameters:

bankDataId `<string>` - id of the bank account details assigned by marketplace

verificationId `<long>` – id of verification in context of which we provide data. Always required.

accountNumberRequested `<string>` – bank account number declared for withdrawals by submerchant [max 32 characters, IBAN format without spaces]

hasDocument `<boolean as string>` – true/false – flag informing that this bank account for payouts should be verified on base of bank document. It must be set to “true” when the method used is “BANK_STATEMENT”

swiftCode `<string>` – standard format of Bank Identifier Code [Max 16 characters]

payoutDataVerificationType `<string>` – type of verification used for the provided payout details verification. Different for each verification type

LIAN_LIAN – special type for Chinese submerchants

PINGPONG - special type for Chinese submerchants

OFX - special type for Chinese submerchants

BANK_STATEMENT – used if all data are present on the document (hasDocument must be “true”)

BANK_TRANSFER

NO_VERIFICATION_NEEDED – used if the provided bank account number was not yet verified by 3rd party but it doesn't have to be verified at the time of providing the bank details. It is used for the occasional transaction process when the bank account number is provided at the beginning of the process.

WORLDFIRST – for worldfirst accounts.

foreign `<boolean as string>` – yes/no - indicator for the foreign transfer, if it's domestic parameter must be set to “no”.

verificationTransferId `<string>` – id of the verification transfer received in the response to the POST /verification-transfers/manual.

verified `<boolean as string>` – parameter indicating whether the bank account number was already verified e.g. by another provider. If set to “false” and the verification type is other than “NO_VERIFICATION_NEEDED” PayU must verify the bank account details.

bankName `<string>` – required for foreign bank account, name of the bank [max 128 characters]

bankCountry `<string>` – required for foreign bank account, bank country [max 2 characters]

bankAddress `<string>` – bank address [max 128 characters]

statementData – details from bank statement

accountNumberFromBank `<string>` – required, bank account number from bank transfer

ownerName `<string>` – required, name of the account owner from bank transfer

address `<string>` - required

street Street with building and apartment number

`<string>` [1, 255 characters]

Zipcode [1, 20 characters]

`<string>`

city [1, 40 characters]

`<string>`

country string (two letter ISO country code)

`<string>` [1, 80 characters]

Field	Verification transfer service	Bank transfer w/o VTS	Bank statement	Comment
bankDataId	R	R	R	
verificationId	R	R	R	
accountNumberRequested	R	R	R	
hasDocument	NA	NA	R	

swiftCode	R	O/R	O/R	Required for foreign transfer
payoutDataVerificationType	R	R	R	
bankName	O	O/R	O/R	Required for foreign transfer
bankAddress	O	O/R	O/R	Required for foreign transfer
bankCountry	O	O/R	O/R	Required for foreign transfer
statementData	O	O	O	
accountNumberFromBank	O/R	O/R	O/R	Depends on the agreement with PayU
ownerName	O/R	O/R	O/R	
address	O/R	O/R	O/R	
street	O/R	O/R	O/R	
zipcode	O/R	O/R	O/R	
city	O/R	O/R	O/R	
country	O/R	O/R	O/R	
foreign	R	R	R	
verified	R	R	R	
verificationTransferId	R	NA	NA	

R - required

O - optional

O/R – optional or required, depending on the agreement with PayU

NA - not applicable

Error responses:

In case of invalid account number, PayU returns error:

```
{
  "message": "Bank Account Number is invalid",
  "code": "409"
}
```

3.6.1. Verification transfer service

This is the recommended method for bank account verification.



After submitting a request to create a verification transfer (/verification-transfers/manual), you have **72 hours** to provide the details of the completed verification transfer using the endpoint /dataloading/payouts/bankAccountData.

If the data is not submitted within this timeframe, the entire process must be repeated.

In order to verify payouts data with use of verification transfer service there is a need to follow the steps:

1. Submerchant must declare the bank account number used for withdrawals
2. Send the request to create order for verification transfer:

POST [/api/aml-verification/v1/verification-transfers/manual](#)

```
{  
  "verificationId": "verification-123",  
  "transferCurrency": "PLN",  
  "email": "user@example.com",  
  "language": "it"  
}
```

Parameters details:

language `<string>` - optional parameter that changes the language of the sent e-mail message.

Response contains details for bank transfer:

http 201

```
{  
  "verificationTransferId": "5f7007da-8e05-4f0f-9503-05c053034bde",  
  "expirationTime": "2020-08-20T14:03:44.473Z",  
  "transferData": {  
    "bankName": "mBank S.A.",  
    "accountNumber": "PL12114011243630011718031748",  
    "swiftCode": "BREXPLPW",  
    "recipientName": "PayU S.A.",  
    "recipientAddress": "ul. Grunwaldzka 186, 60-166 Poznań",  
    "transferTitle": "Verification transfer XX1718031748XX",  
    "transferAmount": 101,  
    "transferCurrency": "PLN"  
  }  
}
```

3. Submerchant needs to make bank transfer from its own bank account to bank account returned as a response on the create verification transfer request. Bank transfer needs to be delivered on PayU bank account before indicated expiration time [expirationTime].
4. Send the request with declarative data and the verification transfer id:

```
POST /api/aml-verification/v1/dataloading/payouts/bankAccountData
```

```
{
  "bankDataId": "01102020",
  "verificationId": "verification-123",
  "accountNumberRequested": "PL1122334455667788991010111223345",
  "hasDocument": "false",
  "payoutDataVerificationType": "BANK_TRANSFER",
  "foreign": "No",
  "verificationTransferId": "5f7007da-8e05-4f0f-9503-05c053034bde",
  "verified": "false"
}
```

verificationTransferId – needs to be indicated with request with declared payouts data (Point 3.6.1 Penny transfer - Example with Verification transfer service). Payouts data need to be declared in 72 hours from above request/response.

Other http responses with errors:

- 400 - Error encountered during posting the data for verification.
- 403 - Error encountered during posting the data for verification.
- 404 - Error encountered during posting the data for verification.
- 409- Verification state does not allow operation.

3.6.2. Bank transfer without verification transfer service

Transfer of the small amount of money done outside the verification transfer service.

```
POST /api/aml-verification/v1/dataloading/payouts/bankAccountData
```

Example request for domestic transfer:

```
{
  "bankDataId": "acc1sd",
  "verificationId": "das5jda",
  "accountNumberRequested": "PL7391739127391273",
  "hasDocument": "false",
  "bankCountry": "PL",
  "payoutDataVerificationType": "BANK_TRANSFER",
  "statementData": {
    "accountNumberFromBank": "PL7391739127391273",
    "ownerName": "Marian Bułka",
    "address": {
      "street": "ul. Nowy Świat 123",
      "city": "Warszawa",
      "zip": "00-100"
    }
  }
}
```

```

    "street": "Al. Jerozolimskie",
    "zipcode": "60-000",
    "city": "Warszawa",
    "country": "PL"
},
"foreign": "no",
"verified": "false"
}

```

Example request for foreign transfer:

```

{
"bankDataId": "acc1sd",
"verificationId": "das5jda",
"accountNumberRequested": "DE7391739127391273",
"hasDocument": "false",
"swiftCode": "BREXPLPW",
"bankAddress": " Schule Str. 23 60-000 Berlin",
"bankCountry": "DE",
"payoutDataVerificationType": "BANK_TRANSFER",
"statementData": {
    "accountNumberFromBank": "DE84074749817492314",
    "ownerName": "Marian Brotchen",
    "address": {
        "street": "Schule Str. 23",
        "zipcode": "60-000",
        "city": "Berlin",
        "country": "DE"
    },
    "foreign": "yes",
    "verified": "false"
}

```

Responses:

http 204 – Success
 http 400 – Invalid input. Reason specified in content
 http 404 – Verification not found
 http 409 – Verification state does not allow operation

3.6.3. Bank statement

If the details from bank for comparison of the bank account data are provided on the document, method used must be “BANK_STATEMENT”. Sample request:

POST [/api/aml-verification/v1/dataloading/payouts/bankAccountData](#)

```
{

```



```
        "bankDataId": "01102020",
        "verificationId": "verification-123",
        "accountNumberRequested": "PL1122334455667788991010111223345",
        "hasDocument": "true",
        "payoutDataVerificationType": "BANK_STATEMENT",
        "foreign": "No",
        "verified": "false"
    }
```

There is also needed to upload document and indicate its type as 'BANK_ACCOUNT AGREEMENT'.

3.6.4. Payoneer

Method is reserved for the users who have verified Payoneer account.

POST [/api/aml-verification/v1/dataloading/payouts/payoneer](#)

```
{
    "bankDataId": "acc1sd",
    "verificationId": "hfs6sfa",
    "payoneerId": "string"
}
```

payoneerId `<string>` – id of the account of submerchant in Payoneer.

Responses:

http 204 – Success.
http 400 – Invalid input. Reason specified in content.
http 404 – Verification not found.
http 409 – Verification state does not allow operation.

3.6.5. Worldfirst

Method is reserved for the users who have verified account in the Worldfirst system.

POST [/api/aml-verification/v1/dataloading/payouts/worldfirst](#)

```
{
    "bankDataId": "acc1sd",
    "verificationId": "5002z000004UgA3AAK",
    "referenceCustomerId": "string"
}
```

referenceCustomerId `<string>` – id of account binding between marketplace and WorldFirst.



Responses:

http 204 - Success.
http 400 – Invalid input. Reason specified in content.
http 404 – Verification not found.
http 409 – Verification state does not allow operation.

3.7. Notification that collecting data and documents has been finished

If all necessary data and documents were provided by submerchant, request to inform PayU about completing this step of a process is required:

POST [/api/aml-verification/v1/verification/complete](#)

```
{  
  "verificationId": "hfs6sfa"  
}
```

Responses:

http 204 - Success.
http 400 – Invalid input. Reason specified in content.
http 404 – Verification not found.
http 409 – Verification state does not allow operation.

If seller did not accept Terms&Conditions PayU will respond with error:

```
{  
  "code": 400,  
  "message": "Terms and Conditions have not been accepted."  
}
```

This error code means that marketplace does not use PayU's T&C widget. There is a possibility to disable T&C validation after contacting PayU.

3.8. Cancellation

Marketplace can cancel verification on every step of verification providing the reason.

POST [/api/aml-verification/v1/verification/cancel](#)

```
{  
  "verificationId": "{{verificationID}}",  
  "rejectionReasonComment": "cancellation reason"  
}
```

Responses:

http 204 - Success.
http 404 – Verification not found.

http 409 – Verification state does not allow cancel.

3.9. Getting status

3.9.1. Status change notification

PayU notifies merchant, on indicated endpoint, about change of status.

```
{
  "verificationType": "FULL",
  "verificationTransferStatus": "Matched",
  "verificationId": "5002500000BVmycAAD",
  "updateDate": "2020-04-03T09:14:06.000Z",
  "status": "REJECTED",
  "processType": "OCCASIONAL",
  "sellerId": "123456test12",
  "requirements": [],
  "requirementNote": null,
  "note": null,
  "errors": [
    {
      "content": {
        "itemType": "ASSOCIATE_DOCUMENT",
        "itemProperties": {
          "documentType": "PEP_STATEMENT",
          "documentId": null,
          "associateType": "BENEFICIARY",
          "associateSurname": "KOWALSKI",
          "associateName": "JAN",
          "associateId": "a2L3L0000008QvXUAU"
        }
      },
      "code": "015"
    }
  ],
  "createDate": "2020-04-03T06:36:15.000Z"
}
```

This notification is also send for initialization of verification done by PayU.

After sending the notification, PayU system requires 200 HTTP status code in the response. Otherwise it will send the notification again. Your system should also take into account the situations when the notification is sent more than once.

3.9.2. Notification Renewal

Notification renewal process is responsible for resending unsuccessful notifications. When the notification is not received, it is sent again after a specified period of time. The maximum



number of retries for a single request is 16, and the waiting time between each retry depends on the number of retries attempted for the notification that was sent.

Resend time interval can be modified in the Custom Setting in AML Verification & Marketplace Settings.

Interval times are as follows:

#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Time(h)	0	0.25	1	1	1	1	1	3.75	3	3	3	3	3	12	12	12	12

The time is measured in hours, for example 0.25 would translate to 15 minutes. Once the notification is successfully sent and retrieved, the process ends.

3.9.3. Obtaining status

In order get the verification status, please request GET on endpoint indicated below with verificationId parameter in url:

GET </api/aml-verification/v1/verification?id={{id}}>

id = verificationId

/verification?id=123

Response:

- http 200 – Found verification
- http 404 – Verification not found.

http 200 will return body which will contain one of below statuses:

POSITIVE

NEGATIVE

REJECTED

WAITING_FOR_DATA

WAITING_FOR_VERIFICATION

CANCELED

3.9.3.1. Positive

Final status – positive: indicates that Seller is accepted and can process transactions with PayU.

{

```

    "verificationType": "PAYOUT_ACCOUNT_DATA",
    "verificationTransferStatus": "Matched",
    "processType": "OCCASIONAL",
    "verificationId": "5002500000BVmycAAD",
    "updateDate": "2020-04-03T09:14:06.000Z",
    "status": "POSITIVE",
    "sellerId": "123456test12",
    "note": null,
    "errors": null,
    "createDate": "2020-04-03T06:36:15.000Z"
}

```

processType `string` – type of the process, OCCASIONAL or REGULAR values.
verificationTransferStatus `string` – indicates status of the verification transfer, possible values are:

- **Success** – transfer was received by PayU.
- **Failure** – PayU did not receive funds.
- **Matched** – transfer has been matched properly.
- **null** – PayU did not have enough information about the transfer.

3.9.3.2. Negative

Final status – negative: indicates that Seller is rejected and can't process transactions with PayU. There is an error code always send for negative scenario.

```

{
    "verificationType": "FULL",
    "verificationTransferStatus": "null",
    "processType": "OCCASIONAL",
    "verificationId": "5002500000BVmycAAD",
    "updateDate": "2020-04-03T09:14:06.000Z",
    "status": "NEGATIVE",
    "sellerId": "123456test12",
    "note": null,
    "errors": [
        {
            "content": {
                "itemType": "ASSOCIATE",
                "itemProperties": {
                    "documentType": null,
                    "documentId": null,
                    "associateId": "1"
                }
            },
            "code": "017"
        }
    ],
    "createDate": "2020-04-03T06:36:15.000Z"
}

```

3.9.3.3. Rejected

Temporary status – rejected: indicates that provided data must be corrected.

When obtained status is Rejected and PayU provided specific errors, indicated in the notification data or documents must be corrected and delivered the same way as in case of new verification followed by the end request /complete. Only corrected data can be send (there is no need to send everything from the beginning).

```
{
  "verificationType": "FULL",
  "verificationTransferStatus": "null",
  "processType": "OCCASIONAL",
  "verificationId": "5002500000BVmycAAD",
  "updateDate": "2020-04-03T09:14:06.000Z",
  "status": "REJECTED",
  "sellerId": "123456test12",
  "note": null,
  "errors": [
    {
      "content": {
        "itemType": "SELLER_DOCUMENT",
        "itemProperties": {
          "documentType": "PEP_STATEMENT",
          "documentId": "2",
          "associateId": null
        }
      },
      "code": "006"
    }
  ],
  "createDate": "2020-04-03T06:36:15.000Z"
}
```

content – group of fields which indicate which data/document error code concerns

itemType – [SELLER, SELLER_DOCUMENT, ASSOCIATE, ASSOCIATE_DOCUMENT, PAYOUT_ACCOUNT]

associateId

associateName

associateSurname

documentId

documentType - [ID_CARD, PASSPORT, DRIVING_LICENCE, RESIDENCE_PERMIT, OTHER]

sellerDocumentType - [REGISTRY_DOCUMENT, BANK_ACCOUNT AGREEMENT, INVOICE, UBO_STATEMENT, PEP_STATEMENT, CIVIL_LAW AGREEMENT, PROXY_DOCUMENT, OTHER_DOCUMENT]

Code – Error Code listed in section – “Verification statuses and error codes”.

3.9.3.4. Waiting for data

Temporary status – waiting for data is sent in the notification when the verification was initialized but the request to start verification was not yet received. Provide necessary data and documents according to the type of submerchant and request POST on /complete endpoint.

```
{
  "verificationId": "7bfd5",
  "verificationType": "FULL",
  "verificationTransferStatus": "null",
  "sellerId": "s123456",
  "processType": "REGULAR",
  "updateDate": "2019-11-12T14:43:43.976Z",
  "createDate": "2019-11-12T14:43:43.976Z",
  "status": "WAITING_FOR_DATA"
}
```

3.9.3.5. Waiting for verification

Temporary status – waiting for verification informs that data and document and data provided to PayU will be verified by PayU or are currently verified. Wait for notification about status change.

```
{
  "verificationId": "7bfd5",
  "verificationType": "FULL",
  "verificationTransferStatus": "null",
  "sellerId": "s12345",
  "processType": "REGULAR",

  "updateDate": "2019-11-12T14:43:43.976Z",
  "createDate": "2019-11-12T14:43:43.976Z",
  "status": "WAITING_FOR_VERIFICATION"
}
```

3.9.3.6. Cancelled

Final status – canceled. Initiate new verification when it is necessary. Marketplace can not cancel verification initialized by PayU or after sending complete information.

```
{
```

```

    "verificationId": "7bfd5",
    "verificationType": "FULL",
    "verificationTransferStatus": "null",
    "sellerId": "s12345",
    "processType": "REGULAR",
    "updateDate": "2019-11-12T14:43:43.976Z",
    "createDate": "2019-11-12T14:43:43.976Z",
    "status": "CANCELLED"
}

```

3.10. Retrieving Sellers Registry Data

You can retrieve registry data for specific seller.

The Retrieving Sellers Registry Data service requires additional configuration from PayU, along with specific requirements that the marketplace platform must meet. Before integration, please contact PayU to confirm your platform's compatibility with this service.

Endpoint:

GET `/api/aml-verification/v1/sellers/{{SELLER_ID}}/registryData`

SELLER_ID (required) - Seller identifier, assigned by marketplace.

Responses:

Http 200:

```
{
  "taxId" : "123",
  "registrationNumber" : "123",
  "personalIdentificationNumber" : null,
  "name" : "ASD Trader",
  "legalForm" : "Limited liability company",
  "formOfMerchantRepresentation" : "KAŻDY ZE WSPÓLNIKÓW REPREZENTUJE SPÓŁKĘ
SAMODZIELNIE",
  "documents" : [
    {
      "type" : "REGISTRY_DOCUMENT",
      "issueDate" : null,
      "files" : [ "a3RPu0000003VGzMAM" ],
      "expireDate" : null,
      "documentNumber" : null
    }
  ],
  "dateOfIncorporation" : "2024-04-01",
  "countryOfIncorporation" : "PL",
  "businessActivityStatus" : "Active",
  "associates" : [
    {
      "name" : "ASD Sp. z o.o.",
      "role" : "OWNER"
    }
  ]
}
```

```

{
  "secondDocumentType" : null,
  "secondDocumentNumber" : null,
  "secondDocumentExpiryDate" : null,
  "nationality" : "PL",
  "documentType" : null,
  "documents" : [
    {
      "type" : "ID_CARD",
      "issueDate" : null,
      "files" : [ "068Pu000008ptltIAA", "068Pu000008pxsiIAA" ],
      "expireDate" : null,
      "documentNumber" : null
    }
  ],
  "documentNumber" : null,
  "documentExpiryDate" : null,
  "associateSurname" : "Surname",
  "associateName" : "Test",
  "associateIdentityNumber" : "94091100009",
  "associateCitizenship" : "PL",
  "associateBirthDate" : "2004-04-17"
  }
],
"address" : {
  "zipcode" : "61-896",
  "street" : "ul. baraniaka 2/112",
  "country" : "PL",
  "city" : "swadzim"
},
"activePayoutDatas" : [
  {
    "verificationTransferId" : null,
    "verificationId" : null,
    "swiftCode" : null,
    "statementData" : {
      "ownerName" : "BrainLess",
      "addressRaw" : "13 M.162 01-793 WARSZAWA",
      "address" : {
        "zipcode" : null,
        "street" : null,
        "country" : null,
        "city" : null
      },
      "accountNumberFromBank" : "PL49105010251000009228788007"
    },
    "referenceCustomerId" : null,
    "payoutDataVerificationType" : "BANK_TRANSFER",
    "payoneerId" : null,
    "foreign" : "no",
  }
]
}

```

```
        "bankName" : null,
        "bankCountry" : "PL",
        "bankAddress" : null,
        "accountNumberRequested" : "PL49105010251000009228788007"
    }
]
}
```

Http 400:

```
{
  "status" : "Error",
  "message" : "There is no registry data for this seller"
}
```

Http 404:

```
{
  "status" : "Error",
  "message" : "No sellers with specified Seller Id found"
}
```

Http 500:

```
{
  "status" : "Error",
  "message" : "Endpoint disabled"
}
```

3.11. Multi-accounting

Multi-accounting in PayU enables sellers to associate multiple bank accounts with a single **sellerId**. This feature is particularly beneficial for sellers who offer their goods on a single marketplace but across multiple markets that use different currencies. With multi-accounting, these sellers can have separate wallets within PayU and easily withdraw funds to the respective bank accounts associated with each market.

The Multi-Accounting service requires additional configuration from PayU, alongside specific requirements that the marketplace platform must meet. Prior to integration, please reach out to PayU to confirm compatibility of your platform with the Multi-Accounting service.

3.11.1. Adding Additional Bank Accounts for Specific sellerId



If a seller utilizes payoneer or worldfirst accounts, they are restricted to having only a single account. If they desire to transition to another account, they must commence a new verification procedure and substitute the existing account with the new one.

To add a new account for an existing **sellerId**, send a verification request with the **type: UPDATE** and the **overridePayouts** parameter with the value **false**.

POST [/api/aml-verification/v1/verification](#)

```
{  
  "sellerId": "{{SELLER_ID}}",  
  "type": "UPDATE",  
  "overridePayouts" : false  
}
```

overridePayouts – this parameter set to **false** indicates the process of adding another bank account. The number of bank accounts for sellerId is defined in the marketplace configuration on the PayU side. If the number of allowed accounts is exceeded, PayU will return an error.

Sending **overridePayouts : false** is only possible with the **UPDATE** verification type. If you omit **overridePayouts**, this parameter defaults to **true**.

Responses for this request are as follows:

Http 200 – Success

Errors:

When adding new bank accounts, the **type** parameter must be sent with the value **UPDATE**, otherwise you will get an error:

Http 409 - **overridePayouts** option cannot be disabled for this **verificationType**

Additionally, in case you have already reached the limit of active payout accounts for the seller, after completing the verification, you will get the error:

Http 409 - Limit of Active Bank Accounts exceeded

3.11.2. Adding Bank Account Details

After initializing the verification of the new bank account, you need to send its details for PayU. You can find description how to do it in section [3.6. Payouts details](#).

3.11.3. Completing Verification

After you verify the bank account you can complete the verification process. You will find details on how to do it in section [3.7 Notification that collecting data and documents has been finished](#).

3.11.4. Removing Bank Account

If you want to remove one of the seller's bank accounts, you must call the endpoint, and specify the **bankAccountNumber** that you want to remove:

POST [/api/aml-verification/v1/sellers/<EXT_CUSTOMER_ID>/payouts/bankAccount/remove](#)

EXT_CUSTOMER_ID – id of the submerchant from which you want to remove an account.

```
{
  "bankAccountNumber": "PL7910501025100009228788007"
}
```

This endpoint is only used to delete bank accounts with additional validation. It is not possible to delete a bank account if there is only one active bank account for given seller.

Responses:

204 – Success

404 – Seller not found

409 – Last Active Payout cannot be deleted

3.12. Re-verification

PayU on base of risk assessment, to be compliant with law requirements, initiates periodical reverification of sellers. Reverification can be either full (type full), which means that seller needs to deliver all the data, which is the same as in boarding of a new merchant of the same type (legal form) or partial (type reverification) when PayU indicates which documents or data is required. It can be for associate – ID document, because of expiration or excerpt from commercial registry.

PayU requests POST on merchant's endpoint:

```
{
  "verificationType": "REVERIFICATION",
  "verificationTransferStatus": "Matched",
  "verificationId": "5008e000009HUuOAAW",
  "updateDate": "2023-01-09T11:06:18.000Z",
  "status": "WAITING_FOR_DATA",
  "sellerId": "60221537",
  "processType": "REGULAR",
  "payoutBlockadeDate": "2022-11-27",
  "note": "",
  "errors": [
    {
      "content": {
        "code": "REVERIFICATION_FAILED",
        "message": "Verification failed. Please provide valid ID document or extract from commercial registry."}
```

```

    "itemType": "ASSOCIATE_DOCUMENT",
    "itemProperties": {
        "documentType": "PEP_STATEMENT",
        "documentId": null,
        "associateType": "BENEFICIARY",
        "associateSurname": "KOWALSKI",
        "associateName": "JAN",
        "associateId": "a2L3L0000008QvXUAU"
    }
},
"code": "015"
},
{
    "content": {
        "itemType": "SELLER_DOCUMENT",
        "itemProperties": {
            "documentType": "PEP_STATEMENT",
            "documentId": "2",
            "associateId": "null"
        }
    },
    "code": "006"
}
],
"duplicateSellers": [],
"createDate": "2023-01-09T11:06:18.000Z",
}
}

```

Process is always started by the PayU. Marketplaces cannot initiate or cancel a reverification. If the marketplace attempts to cancel a reverification, the following response will be returned:

```
{
    "message": "Verification state does not allow operation",
    "code": 409
}
```

If the marketplace tries to initiate a new reverification, the system will return:

```
{
    "message": "Reverification cannot be initialized",
    "code": 409
}
```

The marketplace is allowed to open a cession case (PERSONAL_ID_TAX_ID_CHANGE or CONVERSION) even if a reverification is currently open. In this case, the existing reverification will be automatically rejected, and a new cession case will be opened - regardless of the TAX ID provided.

3.13. ID Replacement process flow

PayU notifies the merchant when associated IDs are nearing expiration. After the initial notification is sent, subsequent reminders are automatically resent twice: 30 and 60 days after the first notice.

Example of ID Expiry Notification:

```
{
  "verificationType": "ID REPLACEMENT",
  "verificationId": "500Pu000006B9HWIA0",
  "updateDate": "2024-05-20T09:45:05.000Z",
  "status": "WAITING_FOR_DATA",
  "sellerId": "SELLER/26-02-2024/270",
  "processType": "REGULAR",
  "payoutBlockadeDate": null,
  "note": "",
  "errors": [
    {
      "content": {
        "itemType": "ASSOCIATE_DOCUMENT",
        "itemProperties": {
          "documentType": "ID_CARD",
          "associateSurname": "Kowalski",
          "associateName": "Marek",
          "associateId": "a2LPu0000011TKMMA2"
        }
      },
      "code": "015"
    }
  ],
  "duplicateSellers": [],
  "createDate": "2024-05-20T09:45:05.000Z"
}
```

In order to add new document you have to:

1. Resend associate information with the POST request to the `/api/aml-verification/v1/dataloading/associates` endpoint. – [3.4. Associates Data](#)
2. Send new document information with the POST request to the `/api/aml-verification/v1/dataloading/associates/documents` endpoint. – [3.5.2.2. Associates' documents](#)
3. You can also check the verification status with the GET request to the `/api/aml-verification/v1/verification/?id={{VERIFICATION_ID}}` endpoint. – [3.9. Getting Status](#)

3.14. Offboarding

Offboarding is a process of seller account deactivation from the marketplace due to various reasons which can be initialized by Marketplace or by PayU.

3.14.1. Offboarding initialized by Marketplace

To initialize offboarding process, you must call an endpoint a specify **sellerId** which you want to offboard.

POST [/api/aml-verification/v1/verification](#)

```
{  
  "sellerId": "{{SELLER_ID}}",  
  "type": "OFFBOARDING"  
}
```

Response status: 200 on success

Response Body:

```
{  
  "verificationId": "{{VERIFICATION_ID}}",  
  "status": "{{VERIFICATION_STATUS}}",  
  "sellerId": "{{SELLER_ID}}"  
}
```

After finalizing the offboarding process PayU sends notification to Marketplace with process status.

Example notification:

```
{  
  "verificationType": "OFFBOARDING",  
  "verificationId": "{{VERIFICATION_ID}}",  
  "updateDate": "2024-08-06T09:50:03.000Z",  
  "status": "POSITIVE",  
  "sellerId": "{{SELLER_ID}}",  
  "processType": "REGULAR",  
  "createDate": "2024-08-06T09:45:03.000Z",  
  "offboarding": {  
    "contractTerminationDate": "2024-08-06",  
    "reason": "{{OFFBOARDING_REASON}}",  
    "offboardingStatus": "POSITIVE",  
    "payoutCompleted": true  
  }  
}
```

3.14.2. Offboarding initialized by PayU

Offboarding process can be initialized by PayU. Offboarding process starting reason will be indicated in the reason field. Possible values are:

- Reverification
- ID replacement
- Fraud
- Inactive seller

Offboarding process steps for finalized offboarding are as follows:

1. Offboarding started:
 - sending notification about offboarding in progress and reason of offboarding
 - sending an e-mail to the seller with the contract termination
2. Waiting termination period for required data/documents from seller (for offboarding due failed reverification or ID replacement) or new transaction in case of offboarding due inactivity
3. After the notice period has expired – no data send from seller:
 - Creating payouts
 - Deactivation of the wallet
4. Sending notification about positive offboarding

The notification sent at the beginning of the process differs depending on the reason. If the offboarding is not a result of the failed reverification or ID replacement process, you will receive following notification for offboarding initialization:

Example notification:

```
{
  "verificationType": "OFFBOARDING",
  "verificationId": "{{VERIFICATION_ID}}",
  "updateDate": "2024-08-06T09:45:03.000Z",
  "status": "IN_PROGRESS",
  "sellerId": "{{SELLER_ID}}",
  "processType": "REGULAR",
  "offboarding" : {
    "contractTerminationDate": "2026-02-06",
    "reason": "{{OFFBOARDING_REASON}}",
    "payoutCompleted": false,
    "offboardingStatus" : "IN_PROGRESS"
  },
  "createDate": "2025-12-06T09:45:03.000Z"
}
```

Offboarding due to inactivity will start:

- after 14 months of inactivity if there should be reverification or ID replacement processes started

- after 18 months in other cases

Before offboarding due to inactivity for 18 months, after 16 months of inactivity there will be an email send directly to the seller informing that if there will be no activity (no new transactions) in the next 2 months, PayU will start offboarding process.

When offboarding is the result of the unsuccessful reverification or ID replacement, you will receive following notification:

```
{
  "verificationType": "REVERIFICATION",
  "verificationId": "{{VERIFICATION_ID}}",
  "updateDate": "2024-08-06T09:50:03.000Z",
  "status": "NEGATIVE",
  "sellerId": "{{SELLER_ID}}",
  "processType": "REGULAR",
  "createDate": "2024-08-06T09:45:03.000Z",
  "offboarding": {
    "contractTerminationDate": "2024-08-06",
    "reason": "{{OFFBOARDING_REASON}}",
    "offboardingStatus": "IN_PROGRESS",
    "payoutCompleted": false
  },
  "note": "",
  "errors": [
    {
      "content": {
        "itemType": "ASSOCIATE_DOCUMENT",
        "itemProperties": {
          "documentType": "ID_CARD",
          "associateSurname": "Kowalski",
          "associateName": "Marek",
          "associateId": "a2LPu0000011TKMMA2"
        },
        "code": "015"
      }
    }
  ]
}
```

The seller must provide updated versions of the required documents for the reverification process. The process of uploading the necessary documents and data remains unchanged, as described in earlier sections.

Once the necessary documents are provided, the offboarding process will be completed, and the contract termination will be withdrawn. In other cases the contract will be terminated and the seller offboarded.

After submitting all required documents and data for verification (while offboarding is already in progress), you must [complete verification](#), just as in the regular verification/reverification process.

It is not possible to cancel the offboarding process. A **409** response will be returned for any cancellation attempt.

4. Verification statuses and error codes

If verification status is REJECTED or NEGATIVE there are also Error Codes send in the notification, which indicate reason of such status.

Status	Error code	Additional info
Positive		Final status – merchant can process transactions
Negative	<p>'009' – Company Inactive (suspended)</p> <p>'017' – Person rejected by PayU Board</p> <p>'018' - Person is on sanctions list</p> <p>'019' – Seller is on sanctions list</p> <p>'021' – Unable to verify</p> <p>'022' - Fraud</p> <p>'023' - Seller is on Visa blacklist VMSS</p> <p>'024' - Seller is on MasterCard blacklist MATCH</p> <p>'025' - verification closed – lack of documents</p> <p>'026' - Seller is on the internal blacklist</p> <p>'033' - Lack of verification transfer</p> <p>'034' - Complete not received</p> <p>'035' - Verification cancelled by Marketplace</p> <p>'036' - Re-verification process started</p> <p>'038' – Blacklisted country</p>	<p>Final status – merchant can't process transactions with PayU.</p> <p>Error code 025 can exist with rejected or negative status. Rejected and 025 closes verification, Negative and 025 means termination of agreement</p>

Rejected	<p>'001' - Document unreadable</p> <p>'002' - Document expired</p> <p>'003' - Document and given data don't match</p> <p>'004' - Wrong document</p> <p>'005' - Document not signed</p> <p>'006' - Document not translated</p> <p>'007' – Incomplete document</p> <p>'008' - Missing person</p> <p>'009' - Company Inactive (suspended)</p> <p>'010' - Penny transfer - bank account doesn't match</p> <p>'011' - Penny transfer - name doesn't match</p> <p>'012' - Penny transfer - address doesn't match</p> <p>'013' – Bank country code doesn't match</p> <p>'014' - Payoneer boarding failed</p> <p>'015' - Required</p> <p>'016' - Other (Please read note)</p> <p>'020' – Company type doesn't match</p> <p>'025' – verification closed – lack of documents</p> <p>'027' - Document and given data don't match - wrong address</p> <p>'028' - Document and given data don't match - wrong TaxId or PersonalIdentificationNumber</p> <p>'029' - Document and given data don't match - wrong Company Name</p>	<p>Status which indicates that some data or documents need to be corrected or delivered.</p> <p>ATTENTION</p> <p>Tax identification number can be changed only when code 028 was send to marketplace. Otherwise an error will be returned.</p>
----------	--	---

	'030' - Missing original ID '031' - Missing second ID '032' - Verification transfer failed '037' – WorldFirst boarding failed '039' – Missing country of birth	
WAITING_FOR_DATA		Temporary status - when verification is initiated, but there was no message on /verification/complete yet.
WAITING_FOR_VERIFICATION		Temporary status – when verification by PayU is in progress
CANCELLED		Verification can be cancelled before sending message on /verification/complete

4.1. Payout lock – not implemented yet

In some situations, for example as a result of verification or at beginning of re-verification, PayU can block payouts for indicated submerchant.

PayU will request POST on merchant's endpoint:

```
{
  "payoutBlock": 1,
  "verificationId": 123456,
  "reason": "verification",
  "sellerId": "string"
}
```

payoutBlock – 1 – turn on block; 0 – turn off block

verificationId – will be delivered when block is connected with verification

reason –

- verification
- occasional_limit_exceeded
- fraud
- bailiff

Response:

http 200 – OK

http 400 – Error

http 404 – Seller not found

5. Examples of requests

Below sample flow of requests/responses. If response for request from PayU side was just http 200 it was omitted.

Step	Request/Response/Notification	Notes
1.	POST on /api/aml-verification/v1/verification { "sellerId": "1e2fff343ju89534583", "type": "FULL" }	Marketplace initialize verification indicating type FULL or PAYOUT_ACCOUNT_DATA for occasional sale
2.	Http 200 with body: { "verificationId": "5002500000C7OFEAA3", "status": "WAITING_FOR_DATA", "sellerId": "1e2fff343ju89534583" }	PayU response with Verification ID
4.	POST on /api/aml-verification/v1/dataloading/seller { "verificationId": "5002500000C7OFEAA3", "sellerId": "1e2fff343ju89534583", "taxId": null, "email": "some@email.com", "name": "John", "surname": "Doe", "legalForm": "PRIVATE_PERSON", "personalIdentificationNumber": "22112222333444", "dateOfBirth": "1988-10-11", "phone": "555333444", "address": { "street": "street1", "zipcode": "11-111", "city": "city", "country": "PL" } }	Marketplace provide Seller data.
5.	POST on /api/aml-verification/v1/dataloading/associates { "verificationId": "5002500000C7OFEAA3", "associateId": "1", "associateType": "REPRESENTATIVE", "associateName": "John", "associateSurname": "Doe", "associateBirthDate": "1970-01-01", "associateCitizenship": "RO" }	Marketplace upload associate's data
6.	POST on /api/aml-verification/v1/dataloading/files body=<multipart/form-data> containing filename, content of file and verificationId	Merchant upload file.
7.	http 200 body: { "verificationId": "5002500000C7OFEAA3", "fileId": "0682500000uN8BAAU" }	PayU indicate fileId in response

	}	
8.	POST on /api/aml-verification/v1/dataloading/associates/documents <pre>{ "verificationId": "5002500000C70FEAA3", "associateId": "1", "type": "PASSPORT", "files": ["0682500000tpoBAAQ", "0682500000tsoBCCQ"], "documentId": "1sd3325234fss", "documentNumber": "sdc443342" }</pre>	Marketplace indicate which file is which document.
9.	POST on /api/aml-verification/v1/dataloading/payouts/bankAccountData <pre>{ "bankDataId": "acc1sd", "verificationId": "5002500000C70FEAA3", "payoutDataVerificationType": "BANK_TRANSFER", "accountNumberRequested": "467329847297491", "verificationTransferId": "1122344", "hasDocument": "false", "verified": "false" }</pre>	Marketplace indicates bank account for payouts
10.	POST on /api/aml-verification/v1/verification/complete <pre>{ "verificationId": "5002500000C70FEAA3", }</pre>	Marketplace indicate that data and documents are provided - verification can start.
11.a	<pre>{ "verificationType": "FULL", "verificationId": "5002500000C70FEAA3", "updateDate": "2020-07-28T05:24:44.000Z", "status": "REJECTED", "processType": "REGULAR", "sellerId": "1e2fff343ju89534583", "note": "Testing the other code:", "errors": [{ "content": { "itemType": "PAYOUT_ACCOUNT", "itemProperties": {} }, "code": "010" }, { "content": { "itemType": "SELLER_DOCUMENT", "itemProperties": { "documentType": "PEP_STATEMENT", "documentId": "2" } }, "code": "005" }, { "content": { "itemType": "SELLER_DOCUMENT", "itemProperties": {} } }] }</pre>	PayU notify marketplace on indicated endpoint with results of verification. Can be POSITIVE, NEGATIVE or REJECTED as in this case. Indicated error codes are shown for introduce structure and possibility to contain many of them in response. Next request example will be technically correct, but not concerning those error codes types.

	<pre> "itemProperties": { "documentType": "PEP_STATEMENT", "documentId": "2" }, "code": "006" }, { "content": { "itemType": "SELLER_DOCUMENT", "itemProperties": { "documentType": "PEP_STATEMENT", "documentId": "2" } }, "code": "007" }], "createDate": "2020-07-21T08:23:49.000Z" } </pre>	
11. b	<pre> {"verificationType": "FULL", "verificationId": "5002500000D4EsZAAV", "updateDate": "2021-05-31T11:51:53.000Z", "status": "REJECTED", "sellerId": "31052021", "processType": "REGULAR", "note": "", "errors": [{ "content": { "itemType": "SELLER_DOCUMENT", "itemProperties": { "documentType": "UBO_STATEMENT" } }, "code": "015" }, { "content": { "itemType": "ASSOCIATE", "itemProperties": { "associateType": "BENEFICIARY", "associateSurname": "Kołda", "associateName": "Małgorzata", "associateId": "a2L25000002EnFzEAK" } }, "code": "008" }, { "content": { "itemType": "PAYOUT_ACCOUNT", "itemProperties": {} }, "code": "010" }], "content": { "itemType": "ASSOCIATE_DOCUMENT", "itemProperties": {} } </pre>	

	<pre> "documentType":"ID_CARD", "associateId":"a2L25000002EnFzEAK" }, "code":"002" }], "createDate":"2021-05-31T11:40:48.000Z } </pre>	
11.c	<pre> {"verificationType":"FULL", "verificationId": "5002500000D4EsZAAV", "updateDate":"2021-05-31T12:07:04.000Z", "status":"REJECTED", "sellerId":"31052021", "processType":"REGULAR", "note":"Description here", "errors":[{ "content":{ "itemType":"SELLER_DOCUMENT", "itemProperties":{ "documentType":"UBO_STATEMENT" } }, "code":"015" }, { "content":{ "itemType":"SELLER", "itemProperties":{} }, "code":"016" }, { "content":{ "itemType":"ASSOCIATE_DOCUMENT", "itemProperties":{ "documentType":"PEP_STATEMENT", "associateType":"BENEFICIARY", "associateSurname":"Kołda", "associateName":"Małgorzata", "associateId":"a2L25000002EnFzEAK" } }, "code":"015" }, { "content":{ "itemType":"AYOUT_ACCOUNT", "itemProperties":{} }, "code":"010" }, { "content":{ "itemType":"ASSOCIATE_DOCUMENT", "itemProperties":{ "documentType":"ID_CARD", "associateId":"a2L25000002EnFzEAK" } }, "code":"001" } </pre>	

	<pre> }, { "content": { "itemType": "SELLER_DOCUMENT", "itemProperties": { "documentType": "REGISTRY_DOCUMENT" } }, "code": "003" }], "createDate": "2021-05-31T11:40:48.000Z" } </pre>	
12.	<p>POST on /api/aml-verification/v1/dataloading/seller</p> <pre> { "verificationId": "5002500000C70FEAA3", "sellerId": "1e2fff343ju89534583", "taxId": null, "email": "some@email.com", "name": "John", "surname": "Doe", "legalForm": "PRIVATE_PERSON", "phone": "555333444", "address": { "street": "street1", "zipcode": "11-111", "city": "city", "country": "PL" } } </pre>	In case of REJCETED status Marketplace provide corrected data according to instruction in error codes. For example seller data
13.	<p>POST on /api/aml-verification/v1/verification/complete</p> <pre> { "verificationId": "5002500000C70FEAA3", } </pre>	Marketplace indicate that data and documents are provided and verification can start.
14.	<pre> { "verificationType": "FULL", "processTye": "REGULAR", "verificationId": "5002500000C70FEAA3", "updateDate": "2020-07-28T05:24:44.000Z", "status": "POSITIVE", "sellerId": "1e2fff343ju89534583", "note": null } </pre>	PayU notify Marketplace on indicated endpoint with result of verification.

6. Reference

6.1. Global Parameters

Parameter	Format
verificationId	String(18)
sellerId	String(50)

6.2. Endpoint Specific Parameters

[/api/aml-verification/v1/verification](#)

Section	Parameter	Format
	type	String, possible values: <ul style="list-style-type: none">• FULL,• PAYOUT_ACCOUNT_DATA,• UPDATE,• REVERIFICATION,• CONVERSION,• PERSONAL_ID_TAX_ID_CHANGE.
	overridePayouts	Boolean

[/api/aml-verification/v1/verification/regulations-acceptance](#)

Section	Parameter	Format
	regulationsUrl	String(255)

[/api/aml-verification/v1/dataloading/seller](#)

Section	Parameter	Format
	taxId	String(60)
	personalIdentificationNumber	String(150)
	companyName	String(80)
	name	String(40)
	surname	String(40)
	legalForm	String, possible values: <ul style="list-style-type: none">• PRIVATE_PERSON,• SOLE_TRADER, LEGAL_ENTITY,• ASSOCIATION,• CIVIL_LAW_PARTNERSHIP,• FOREIGN_COMPANY,• FOUNDATION,• GENERAL_PARTNERSHIP,• JOINT_STOCK_COMPANY,• LIMITED_JOINT_STOCK_PARTNERSHIP,• LIMITED LIABILITY COMPANY,• LIMITED LIABILITY PARTNERSHIP,• PROFESSIONAL_PARTNERSHIP,• LIMITED_PARTNERSHIP,• OTHER.
	gusCode	String(50)
	registryNumber	String(40)

	email	String(80)
	phone	String(40)
	isAccountCloned	String, possible values: • true, • false.
	dateOfBirth	String, YYYY-MM-DD
address	street	String(20)
	zipcode	String(40)
	city	String(40)
	country	String(80)
uboStatement	stockExchange	String(255)
	isin	String(12)
	physicalBeneficiary	String, possible values: • true, • false.

</api/aml-verification/v1/dataloading/seller/documents>

	documentId	String(100)
	type	String, possible values: • REGISTRY DOCUMENT, • BANK ACCOUNT AGREEMENT, • UBO STATEMENT, • PEP STATEMENT, • CIVIL LAW AGREEMENT, • PROXY DOCUMENT, • OTHER DOCUMENT, • ID CARD, • INVOICE, • PASSPORT, • DRIVING LICENCE, • RESIDENCE PERMIT, • FUNCTION CONFIRMATION, • ACT OF FOUNDATION, • BUSINESS TERMS, • ACCEPTANCE AGREEMENT, • FURTHER GUIDELINES, • OTHER AGREEMENT, • SELFIE WITH ID, • SALE CESSION AGREEMENT, • NOTARIAL ACT.
	note	String(255)
	files	String[]
	translationFiles	String[]

</api/aml-verification/v1/dataloading/payouts/bankAccountData>

Section	Parameter	Format
bankAccountData	bankDataId	String(100)
	accountNumberRequested	String(255)
	hasDocument	String, possible values: <ul style="list-style-type: none">• true,• false.
	swiftCode	String(11)
	payoutDataVerificationType	String, possible values: <ul style="list-style-type: none">• LIAN_LIAN,• BANK_STATEMENT,• BANK_TRANSFER,• PINGPONG,• NO_VERIFICATION_NEEDED,• OFX.
	foreign	String, possible values: <ul style="list-style-type: none">• yes,• no.
	verified	String, possible values: <ul style="list-style-type: none">• true,• false.
	verificationTransferId	String(255)
	bankName	String(150)
	bankCountry	String(10)
statementData	accountNumberFromBank	String(50)
	ownerName	String(255)
	address	street
		zipcode
		city
		country
	addressRaw	String(255)

</api/aml-verification/v1/dataloading/payouts/payoneer>

Section	Parameter	Format
	bankDataId	String(100)
	payoneerId	String(50)

</api/aml-verification/v1/dataloading/associates>

Section	Parameter	Format
	associateId	String(100)
	associateType	String, possible values: <ul style="list-style-type: none">• BENEFICIARY,

		<ul style="list-style-type: none"> • REPRESENTATIVE.
associateName		String(200)
associateSurname		String(200)
associateCitizenship		String(255)
associateIdentityNumber		String(150)
associateBirthDate		String, YYYY-MM-DD
associateCountryOfBirth		String(255)
pepStatement		String, possible values: <ul style="list-style-type: none"> • true, • false.

[/api/aml-verification/v1/dataloading/associated/documents](#)

Section	Parameter	Format
	documentId	String(100)
	type	String, possible values: <ul style="list-style-type: none"> • REGISTRY DOCUMENT, • BANK ACCOUNT AGREEMENT, • UBO STATEMENT, • PEP STATEMENT, • CIVIL LAW AGREEMENT, • PROXY DOCUMENT, • OTHER DOCUMENT, • ID CARD, • INVOICE, • PASSPORT, • DRIVING LICENCE, • RESIDENCE PERMIT, • FUNCTION CONFIRMATION, • ACT OF FOUNDATION, • BUSINESS TERMS, • ACCEPTANCE AGREEMENT, • FURTHER GUIDELINES, • OTHER AGREEMENT, • SELFIE WITH ID, • SALE CESSION AGREEMENT, • NOTARIAL ACT.
	documentNumber	String(100)
	issueDate	String, YYYY-MM-DD
	expireDate	String, YYYY-MM-DD
	note	String(255)
	files	String[]
	translationFiles	String[]

/api/aml-verification/v1/dataloading/seller/riskData

Section	Parameter	Format
	monthlyTurnover	Double(16,2)
	category	String(255)
	riskClass	String(100)