Automatic testing - Marketplace AML API

1. Verification

Testing automation of the AML API allows to test the process without PayU involvement. The process consists in sending a proper values in the request body, depending on the desired verification result: POSITIVE, REJECTED, NEGATIVE and error codes in case of rejected and negative statuses.

1.1. Simulating errors

If you want to simulate some of the edge cases, you need to set specific error codes as the value of the **gusCode** field at the correct stage of the verification process:

Used errorCode	How to simulate	Verification Step	Result
008	Send **008** in **gusCode** field with seller data + don't send CRP for this verification.	dataloading/seller	Error: 008 – Missing Person
015	Send **015** in **gusCode** field with the seller data.	dataloading/seller	Returned UBO_STATEMENT
015	Send **015** in **gusCode** field with the associates data.	dataloading/associates	Returned PEP_STATEMENT

Automatic testing steps are presented below.

1.2. Verification initialization

POST : {{baseURL}}/verification

```
{

  "sellerId": "{{SellerID}}",

  "type": "{{VerificationType}}"

}
```

1.3. Sending the data with desired result indication. Only in this step it is possible to indicate the final verification result (POSITIVE, REJECTED, NEGATIVE)

Data type	**Endpoint + Required fields**	**Body**

Seller data	POST : {{baseURL}}/dataloading/seller
	Required fields : verificationId, sellerId, address, email, gusCode - Error Codes, registryNumber - Verification Status
		{
		"verificationId": "{{VERIFICATION_ID}}",
		"sellerId": "{{SELLER_ID}}",
		"companyName": "Tatar Company7",
		"taxId": "8618132531",
		"legalForm": "SOLE_TRADER",
		"address": {
		"street": "North of Sunzhuang Village, Changziying Town, Daxing District, Beijing",
		"zipcode": "100000",
		"city": "Beijing",
		"country": "CN"
		},
		"email": "guiji19504@163.com",
		″phone": "+86 151 6026 2966",
		"gusCode": "{{ERROR_CODES}}",
		"registryNumber": "{{VERIFICATION_STATUS}}"
		}
		EXAMPLE:
		{
		"verificationId": "5004L00000B27MPQAZ"
		"sellerId": "12345678",
		"companyName": "Tatar Company7",
		"taxId": "8618132531",
		"legalForm": "SOLE_TRADER",
		"address": {
		"street": "ul. Polna",
		"zipcode": "100000",
		"city": "Poznań",
		"country": "PL"
		},
		"email": "test@test.com",
		″phone": "+86 151 6026 2966",
		"gusCode": "015;016",
		"registryNumber": "REJECTED"
		}

Associates	POST : {{baseURL}}/dataloading/associates
	Required fields : verificationId, associateId, associateType, associateSurname, associateName -
		{
		"verificationId": "{{VERIFICATION_ID}}",
		"associateId": "TestRepresentative1",
		"associateType": "REPRESENTATIVE",
		"associateName": "{{ERROR_CODES}}",
		"associateSurname": "Kowalski",
		"associateCitizenship": "Lodz",
		"associateIdentityNumber": "34234324324",
		"associateBirthDate": "2019-11-21"
		}

	Error Codes	**EXAMPLE:**
		{
		"verificationId": "5004L00000B27MPQAZ",
		"associateId": "TestRepresentative1",
		"associateType": "REPRESENTATIVE",
		"associateName": "030",
		"associateSurname": "Kowalski",
		"associateCitizenship": "Lodz",
		"associateIdentityNumber":
		"34234324324",
		"associateBirthDate": "2019-11-21"
		}

Payout Account Data	POST :
	{{baseURL}}/dataloading/payouts/bankAccountData
	
	Required fields: verificationId, bankDataId, bankName - Error Codes
	
	For Payoneer also : payoneerId
	
	For Non-Payoneer :
	accountNumberRequested, verified
		{
		"verificationId": "{{VERIFICATION_ID}}",
		"bankDataId" : "marketplaceBankDataId",
		"payoneerId" : "payoneerID",
		"accountNumberRequested": "",
		"hasDocument" : "false",
		"swiftCode" : "swiftCode",
		"payoutDataVerificationType" :
		"BANK_TRANSFER",
		"bankName" : "{{ERROR_CODES}}",
		"bankCountry" : "bankCountry",
		"statementData" : {
		"accountNumberFromBank": "",
		"ownerName" : "ownerName",
		"address" : {
		"street" : "street",
		"zipcode" : "zipcode",
		"city" : "city",
		"country" : "PL"
		},
		},
		"foreign" : "false",
		"verified": "false",
		"verificationTransferId" :
		"{{VERIFICATION_TRANSFER_ID}}"
		}
		
		Example:
		{
		"verificationId": "5004L00000B27MPQAZ",
		"bankDataId" : "marketplaceBankDataId",
		"accountNumberRequested":
		"PL66804610704885900358899402"
		"hasDocument" : "false",
		"swiftCode" : "swiftCode",
		"payoutDataVerificationType" :
		"BANK_TRANSFER",
		"bankName" : "011;012",
		"bankCountry" : "PL",
		"statementData" : {
		"accountNumberFromBank":"",
		"ownerName" : "ownerName",
		"address" : {
		"street" : "street",
		"zipcode" : "zipcode",
		"city" : "city",
		"country" : "PL"
		},

		`},`
		`"foreign" : "false",`
		`"verified": "false",`
		`"verificationTransferId" : "2233445566"`
		`}`

Documents	POST : {{baseURL}}`/dataloading/seller/documents`
	Required fields : `verificationId`, `documentId`, `type`, `files`, `documentNumber` - Error Codes
		`{`
		`"verificationId": "{{VERIFICATION_ID}}",`
		`"documentId": "TestDocumentId2",`
		`"type": "ID_CARD",`
		`"documentNumber" : "{{ERROR_CODES}}",`
		`"note": "Test note",`
		`"files": [`
		`"0683G000000AqqQQAS"`
		`],`
		`"translationFiles": []`
		`}`
		Example:
		`{`
		`"verificationId": "5004L00000B27MPQAZ",`
		`"documentId": "TestDocumentId2",`
		`"type": "ID_CARD",`
		`"documentNumber" : "002;003",`
		`"note": "Test note",`
		`"files": [`
		`"0683G000000AqqQQAS"`
		`],`
		`"translationFiles": []`
		`}`

Sending multiple error codes in one request is possible by adding them after ; e.g. 010;011

1.4. Finalizing data upload

POST : {{baseURL}}`/verification/complete`

```
{
  "verificationId" : "{{VERIFICATION_ID}}"
}
```

After sending this request, a proper notification will be send on the notification url provided by marketplace.

2. Reverification

Presented below is the mocked process of reverification, which allows to test if you can receive notification about opened reverification process for given merchant. API test mode will allow you to send **sellerData** and associates with expected **errorCodes**.

As a result:

- Initialized verification will be closed as **POSITIVE**
- Reverification will be created and automatically send to marketplace.

2.1. Verification Initialization

To initialize reverification testing process you should call the : {{baseURL}}/verification endpoint with the **POST** method and request body of:

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

2.2. Sending Seller Data

Next, you need to send the seller's data to {{baseURL}}/dataloading/seller endpoint with **POST** method and request with the following body:

```
{
    "verificationId": "{{VERIFICATION_ID}}",
    "sellerId": "{{SELLER_ID}}",
    "companyName": "Seller Company Name",
    "taxId": "sellerTestTaxId",
    "legalForm": "SOLE_TRADER",
    "gusCode": "022;020",
    "registryNumber": "ID_REPLACEMENT",
    "address":{
        "street": "ul. Grunwaldzka 186",
        "zipcode": "60-166",
        "city": "Poznań",
        "country": "PL"
    },
    "email": "test@payu.com",
    "phone": "+48123456789",
    "dateOfBirth": "1990-12-20"
}
```

Notes:

- **registryNumber** has to be populated with value "REVERIFICATION",
- **gusCode** should be populated with error codes expected to be sent with reverification, codes should be separated with "**;**".

2.3. Sending Associate Data

Now, you need to send the associate data to {{baseURL}}/dataloading/associates endpoint with the **POST** method and the request with the following body:

```
{
  "verificationId": "{{VERIFICATION_ID}}",
  "associateId": "TestRepresentative2",
  "associateType": "REPRESENTATIVE",
  "associateName": "030;031",
  "associateSurname": "Kowalski",
  "associateCitizenship": "PL",
  "associateIdentityNumber": "34234324324",
  "associateBirthDate": "1990-11-21"
}
```

Notes:

- **associateName** should be populated with error codes expected to be sent with reverification. Codes should be separated with ";".

2.4. Completing Test Verification

To complete this step, you need to call the {{baseURL}}/dataloading/associates endpoint with the **POST** method and the request with the following body:

```
{
  "verificationId": "{{VERIFICATION_ID}}"
}
```

Notes:

- After complete operation verification **{{VERIFICATION_ID}}** will be closed as **POSITIVE** and reverification with **{{REVERIFICATION_ID}}** will be created. Corresponding notification will be sent to marketplace.

- Seller **errorCodes** will be sent as provided in the Sending Seller Data step.
- Associates will be copied from **{{VERIFICATION_ID}}** and provided **errorCodes** sent with **{{REVERIFICATION_ID}}.**

2.4.1. Verification Notification

Notification about closed **{{VERIFICATION_ID}}.**

```
{
   "verificationType": "FULL",
   "verificationId": "{{VERIFICATION_ID}}",
   "updateDate": "2023-05-11T13:19:06.000Z",
   "status": "POSITIVE",
   "sellerId": "{{SELLER_ID}}",
   "processType": "REGULAR",
   "note": null,
   "createDate": "2023-05-11T13:12:22.000Z"
}
```

2.4.2. Reverification notification

Notification about newly created Reverification with **{{REVERIFICATION_ID}}.**

```
{
   "verificationType": "REVERIFICATION",
   "verificationId": "{{REVERIFICATION_ID}}",
   "updateDate": "2023-05-11T13:33:29.000Z",
   "status": "WAITING_FOR_DATA",
   "sellerId": "{{SELLER_ID}}",
   "processType": "REGULAR",
   "payoutBlockadeDate": "2023-06-10",
   "note": "Test in Progress | Awaiting for complete",
   "errors": [
      {
         "content": {
            "itemType": "SELLER",
            "itemProperties": {
            }
         },
         "code": "022"
      },
      {
         "content": {
            "itemType": "SELLER",
            "itemProperties": {

            }
         },
         "code": "020"
      },
      {
         "content": {
            "itemType": "ASSOCIATE",
            "itemProperties": {
               "associateSurname": "Kowalski",
               "associateName": "030;031",
               "associateId": "a2L3G000000eDZ0UAM"
            }
         },
         "code": "030"
      },
      {
         "content": {
            "itemType": "ASSOCIATE",
            "itemProperties": {
               "associateSurname": "Kowalski",
               "associateName": "030;031",
               "associateId": "a2L3G000000eDZ0UAM"
            }
         },
         "code": "031"
      }
   ],
   "duplicateSellers": [
      {
         "sellerId": "{{SECOND_SELLER_ID_WITH_THE_SAME_TAX_ID}}"
      }
   ],
   "createDate": "2023-05-11T13:18:58.000Z"
}
```

2.5. Get Test Status

You can retrieve test status for given **{{VERIFICATION_ID}}** by sending a GET request to the {{baseURL}}/verification?id={{VERIFICATION_ID}}

```
{
    "verificationType": "FULL",
    "verificationTransferStatus": null,
    "verificationId": "{{VERIFICATION_ID}}",
    "updateDate": "2023-05-11T13:19:24.000Z",
    "status": "POSITIVE",
    "sellerId": "allegro100520232",
    "processType": "REGULAR",
    "payoutBlockadeDate": null,
    "note": " Test in Progress | ID Replacement notification with verificationId: 
{{IDREVERIFICATION_ID}} has been sent successfully | Awaiting for complete",
    "errors": null,
    "duplicateSellers": null,
    "createDate": "2023-05-11T13:12:22.000Z"
}
```

note parameter can contain following information:

- **Test in Progress | Reverification notification with verificationId: {{REVERIFICATION_ID}} has been sent successfully | Awaiting for complete** – receiving this information means that everything is ok and there is need to send **COMPLETE** to end the test.
- **Test Success** - means that test ended successfully. Complete request has been sent to **{{REVERIFICATION_ID}}**.
- **Test in Progress | Reverification notification with verificationId: {{REVERIFICATION_ID}} has been sent, but error received : Code: 500 Body: REVERIFICATION NOTIFICATION API ERROR**

If the **note** parameter takes the value **Test in Progress**, then to you have to complete a test by sending the Complete Verification request as described in the section 1.4. Without it, it will be impossible to perform next test with the same **{{SELLER_ID}}.**

If the **note** parameter takes the value **Test in Progress** and you send another verification initialization request, then **verificationId : {{REVERIFICATION_ID}}** will be returned, meaning that you must first close the open verification.

2.6. Send additional data

Optionally you can send additional data required in reverification process, such as:

- sellerData
- associates
- seller/documents
- associates/documents

2.7. Reverification complete

To complete the reverification process you need call the {{baseURL}}/verification/complete endpoint with the **POST** method and the following body:

```
{
  "verificationId": "{{REVERIFICATION_ID}}"
}
```

After this step Reverification Case will be closed as POSITIVE, which end the testing process.

3. Id Replacement

The mocked process of ID Replacement verification (IDReverification) allows testing whether the marketplace can receive notifications about the initiation of an ID Replacement process for a given merchant. API Test mode allows to send sellerData and associates, but differently from reverification, no errorCodes are expected, as answer will have 015 error code by default.

As a result:

- Initialized verification will be closed as POSITIVE
- IDReverification will be created and automatically send to marketplace.

3.1. Initialize verification

Method: **POST**

Endpoint: {{baseURL}}/verification

Body:

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

3.2. Send Seller Data

Next you should add the sellers data with registryNumber set to **ID_REPLACEMENT**. Additionally gusCode parameter should be populated with error codes expected to be sent with IDReverification. Codes should be separated with "**;**"

Method: **POST**

Endpoint: {{baseURL}}/dataloading/seller

Body:

```
{
    "verificationId": "{{VERIFICATION_ID}}",
    "sellerId": "{{SELLER_ID}}",
    "companyName": "Seller Company Name",
    "taxId": "sellerTestTaxId",
    "legalForm": "SOLE_TRADER",
    "registryNumber": "ID_REPLACEMENT",
    "address":{
        "street": "ul. Grunwaldzka 186",
        "zipcode": "60-166",
        "city": "Poznań",
        "country": "PL"
    },
    "email": "test@payu.com",
    "phone": "+48123456789",
    "dateOfBirth": "1990-12-20"
}
```

3.3. Send Associate Data

Now, send the associate data:

Method: **POST**

Endpoint: {{baseURL}}/dataloading/associates

Body:

```
{
  "verificationId": "{{VERIFICATION_ID}}",
  "associateId": "TestRepresentative2",
  "associateType": "REPRESENTATIVE",
  "associateName": "Jan",
  "associateSurname": "Kowalski",
  "associateCitizenship": "PL",
  "associateIdentityNumber": "34234324324",
  "associateBirthDate": "1990-11-21"
}
```

3.4. Accept Regulations

Method: **POST**

Endpoint: {{baseURL}} verification/regulations-acceptance

Body:

```
{
  "verificationId": "{{VERIFICATION_ID}}",
  "regulationsUrl":"http://static.payu.com/regulations/marketplace-psd2-
regulation-v2.pdf",
}
```

3.5. Complete Test Verification

Method: **POST**

Endpoint: {{baseURL}}/verification/complete

Body:

```
{
  "verificationId": "{{VERIFICATION_ID}}"
}
```

Notes:

- After complete operation verification **{{VERIFICATION_ID}}** will be closed as **POSITIVE** and IdReverification with **{{IDREVERIFICATION_ID}}** will be created. Corresponding notification will be sent to marketplace.
- Associates will be copied from **{{VERIFICATION_ID}}** and errorCode 015 sent with **{{IDREVERIFICATION_ID}}.**

3.5.1. Verification Notification

Notification about closed **{{VERIFICATION_ID}}.**

```
{
   "verificationType": "FULL",
   "verificationId": "{{VERIFICATION_ID}}",
   "updateDate": "2023-05-11T13:19:06.000Z",
   "status": "POSITIVE",
   "sellerId": "{{SELLER_ID}}",
   "processType": "REGULAR",
   "note": null,
   "createDate": "2023-05-11T13:12:22.000Z"
}
```

3.5.2. IdReverification notification

Notification about newly created IdReverification.

```
{
   "verificationType": "ID_REPLACEMENT",
   "verificationId": "{{IDREVERIFICATION_ID}}",
   "updateDate": "2023-05-11T13:33:29.000Z",
   "status": "WAITING_FOR_DATA",
   "sellerId": "{{SELLER_ID}}",
   "processType": "REGULAR",
   "payoutBlockadeDate": "2023-06-10",
   "note": "Test in Progress | Awaiting for complete",
   "errors": [
      {
         "content": {
            "itemType": "ASSOCIATE_DOCUMENT",
            "itemProperties": {
                "documentType": "ID_CARD",
                "associateSurname": "Kowalski",
                "associateName": "015",
                "associateId": "a2L3G000000eDZ0UAM"
            }
         },
         "code": "015"
      }
   ],
   "createDate": "2023-05-11T13:18:58.000Z"
}
```

3.6. Get Test Status

You can retrieve test status for given **{{VERIFICATION_ID}}** by sending a GET request to the {{baseURL}}/verification?id={{VERIFICATION_ID}}

Method: **GET**

Endpoint: {{baseURL}}/verification?id={{VERIFICATION_ID}}

Body:

```
{
    "verificationType": "FULL",
    "verificationTransferStatus": null,
    "verificationId": "{{VERIFICATION_ID}}",
    "updateDate": "2023-05-11T13:19:24.000Z",
    "status": "POSITIVE",
    "sellerId": "allegro100520232",
    "processType": "REGULAR",
    "payoutBlockadeDate": null,
    "note": " Test in Progress | ID Replacement notification with verificationId:
{{IDREVERIFICATION_ID}} has been sent successfully | Awaiting for complete",
    "errors": null,
    "duplicateSellers": null,
    "createDate": "2023-05-11T13:12:22.000Z"
}
```

note parameter can contain following information:

- **Test in Progress | IdReverification notification with verificationId: {{IDREVERIFICATION_ID}} has been sent successfully | Awaiting for complete** – receiving this information means that everything is ok and there is need to send **COMPLETE** to end the test.
- **Test Success** - means that test ended successfully. Complete request has been sent to **{{IDREVERIFICATION_ID}}**.
- **Test in Progress | IdReverification notification with verificationId: {{IDREVERIFICATION_ID} } has been sent, but error received : Code: 500 Body: REVERIFICATION NOTIFICATION API ERROR**

If the **note** parameter takes the value **Test in Progress**, then to you have to complete a test by sending the Complete Verification request as described in the section 1.4. Without it, it will be impossible to perform next test with the same **{{SELLER_ID}}.**

If the **note** parameter takes the value **Test in Progress** and you send another verification initialization request, then **verificationId : {{IDREVERIFICATION_ID}}** will be returned, meaning that you must first close the open IdReverification.

3.7. Send Additional Data

This step is optional for API Test Mode.

Now marketplace can send additional data required in IdReverification process:

- associates
- associates/documents

3.8. Complete IdReverification

Method: **POST**

Endpoint: {{baseURL}}/verification/complete

Body:

```
{
  "verificationId": "{{IDREVERIFICATION_ID}}"
}
```

After this step Reverification Case will be closed as POSITIVE, which end the testing process.

4. The case of errorCode 008 – Missing Person

If you want to test the flow with the **008 - Missing Person** error, please proceed as follows:

4.1. Initialize verification

Method: **POST**

Endpoint: {{baseURL}}/verification

Body:

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

4.2. Send Seller Data

Method: **POST**

Endpoint: {{baseURL}}/dataloading/seller

Body:

```
{
  "verificationId": "{{VERIFICATION_ID}}",
  "sellerId": "{{SELLER_ID}}",
  "companyName": "Seller Company Name",
  "taxId": "sellerTestTaxId",
  "legalForm": "SOLE_TRADER",
  "gusCode": "008;020",
  "registryNumber": "REJECTED",
  "address": {
    "street": "ul. Grunwaldzka 186",
    "zipcode": "60-166",
    "city": "Poznań",
    "country": "PL"
  },
  "email": "test@payu.com",
  "phone": "+48123456789",
  "dateOfBirth": "1990-12-20"
}
```

Remember that **gusCode** parameter should contain **008**.

4.3. Acepting Regulations

Method: **POST**

Endpoint: {{baseURL}}/verification/regulations-acceptance

Body:

```
{
    "verificationId": "{{VERIFICATION_ID}}",
    "regulationsUrl": "http://static.payu.com/regulations/marketplace-psd2-
regulation-v2.pdf"
}
```

4.4. Completing verification

Method: **POST**

Endpoint: {{baseURL}}/verification/complete

Body:

```
{
    "verificationId": "{{VERIFICATION_ID}}"
}
```

As a result you will receive the notification:

```
{
  "verificationType": "FULL",
  "verificationId": "5008E00000NFC3RQAX",
  "updateDate": "2023-06-27T12:03:12.000Z",
  "status": "REJECTED",
  "sellerId": "{{SELLER_ID}}",
  "processType": "REGULAR",
  "note": "Test Success | Current Status: REJECTED | Expected Status: REJECTED",
  "errors": [
    {
      "content": {
        "itemType": "SELLER",
        "itemProperties": {}
      },
      "code": "008"
    },
    {
      "content": {
        "itemType": "SELLER",
        "itemProperties": {}
      },
      "code": "020"
    },
    {
      "content": {
        "itemType": "ASSOCIATE",
        "itemProperties": {
          "associateSurname": "testLastName",
          "associateName": "008",
          "associateId": "a2L8E000001TJcZUAW"
        }
      },
      "code": "008"
    }
  ],
  "createDate": "2023-06-27T12:02:40.000Z"
}
```

5. Occasional Sale Flow – Limits Exceeded

To simulate a situation where the seller exceeds PayU limits, follow these steps:

5.1. Initialize verification

Method: **POST**

Endpoint: : {{baseURL}}/verification

Body:

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

5.2. Send Seller Data

Next, you need to send the seller's data with registryNumber parameter set to **LIMITS_EXCEEDED**:

Method: **POST**

Endpoint: {{baseURL}}/dataloading/seller

Body:

```
{
  "verificationId": "{{VERIFICATION_ID}}",
  "sellerId": "{{SELLER_ID}}",
  "companyName": "Seller Company Name",
  "taxId": "sellerTestTaxId",
  "legalForm": "SOLE_TRADER",
  "gusCode": "008;020",
  "registryNumber": "LIMITS_EXCEEDED",
  "address": {
    "street": "ul. Grunwaldzka 186",
    "zipcode": "60-166",
    "city": "Poznań",
    "country": "PL"
  },
  "email": "test@payu.com",
  "phone": "+48123456789",
  "dateOfBirth": "1990-12-20"
}
```

5.3. Complete Verification

Method: **POST**

Endpoint: {{baseURL}}/verification/complete

Body:

```
{
    "verificationId": "{{VERIFICATION_ID}}"
}
```