COLLECTING SOLUTION

Marketplace - Onboarding KYC

Document version 1.7

Contents

1. HISTORY OF THE DOCUMENT

Version	Author	Date	Comment
1.7	Lyra Collect	4/13/2022	Document overhaul. Added Single Sign-On authentication mode.
1.6	Lyra Collect	10/6/2021	Update of the *Viewing the status of the enrollment request* chapter.
1.5.1	Lyra Collect	8/25/2021	*Step 1: Creating the seller* chapter: precision in the description of **economic_agent**.
1.5	Lyra Collect	5/27/2021	• Details in the *Resources of the web service*chapter • Clarifications in the chapter *Step 1: Creating the seller* • Addition of *Registering "voucher" contracts* chapter • *Adding, updating or deleting an IBAN* chapter removed
1.4	Lyra Collect	5/5/2021	• Addition of details in the *Resources of the web service*chapter • Update of the *Step 1: Creating the seller* chapter: the "naf_code" code becomes "naf"
1.3	Lyra Collect	4/13/2021	Addition of details in the *Entering the IBANs* chapter.
1.2	Lyra Collect	1/28/2021	Update of the *Step 1: Creating the seller* chapter: • Clarifications added to the attributes vat_number, legal_form, economic_agent and activity.mcc. • Addition of the value list of the legal_form attribute.

1.1	Lyra Collect	12/21/2020	• Clarifications in the chapter *Step 1: Creating the seller.*
			• Addition of chapters on updating the registration (PUT).
			  • *Using the external reference*
			  • *Entering the IBANs*
			  • *Adding, updating or deleting an IBAN*
			  • *Updating a registration*
			  • *Adding or updating a board member*
			  • *Adding, updating or deleting an activity*

1.0	Lyra Collect	1/20/2019	API technical documentation: creation of a seller + KYC document

2. PRESENTATION

This document covers the operations that lead to *vendor enrollment* (resources, associated transactions, authentication, headers).

For other topics (distinguishing between environments, defining webhooks, integrating forms, creating orders, etc.), please refer to the *Marketplace API* Integration Guide.

3. RESOURCES OF THE WEBSERVICE

The Marketplace API uses *Swagger* for displaying the complete list of views.

Depending on the authentication mode that you use (see *Authentication* chapter), the open_api resource is available at:

- **Oauth2 mode:** *https://secure.lyra.com/marketplace/v1/open_api/*
- **Basic mode:** *https://secure.lyra.com/marketplace/open_api/*

4. AUTHENTICATION

Two authentication modes coexist:

- OAuth2: for new Marketplaces created from 2022;
- Basic: used by Marketplaces set up before 2022.

4.1. OAuth2

The OAuth2 mode requires that your administrative contact, i.e. the legal or technical manager of the Marketplace you want to integrate, has opened a technical account for you from their own Merchant Back Office account.

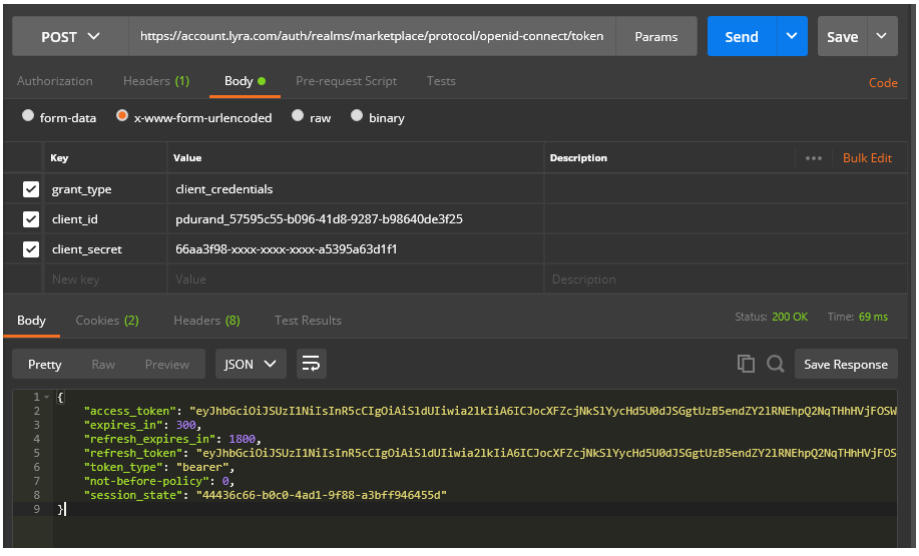
After that:

- Sign in to the Merchant Back Office with your own login details.
- At the top right of the screen, open the drop-down menu under your name, and click **Settings**.
- Click on the **API Key** tab to retrieve your keys for the test and production environments.

To generate an access token, you must make a `POST` request to the following address: *https://account.lyra.com/auth/realms/marketplace/protocol/openid-connect/token*

(same address for both environments) with the following parameters in the `application/x-www-form-urlencoded` format:

- `grant_type`: `client_credentials`
- `client_id`: `CLIENT_ID` (consisting of the first letters of the first and last name, followed by a "_" and the `uuid` of the Marketplace).
- `client_secret`: `CLIENT_SECRET` (corresponding to the “API Key”, a 32-character string in uuid format)



The access token is returned in the `access_token` field.

> The access token has a limited lifetime. You must set up a method for retrieving a new token at no more than every 300 seconds (5 minutes).

A request submitted with an expired authentication token will be rejected with an HTTP 307 code.

The `refresh_token` is not used for the Marketplace.

Example of obtaining the authentication token in PHP

```
<?php

require 'vendor/autoload.php';

use GuzzleHttp\Client;

$MARKETPLACE_UUID = "57595c55-b096-41d8-9287-b98640de3f25";
$USERNAME = 'mdupont';
$CLIENT_ID = $USERNAME."_".$MARKETPLACE_UUID;
$CLIENT_SECRET = "aaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee";

$AUTH_URL = "https://account.lyra.com/auth/realms/marketplace/protocol/openid-connect/";
$API_URL = "https://secure.lyra.com/marketplace-test/v1/";

session_start();

function getAccessToken() {
    global $AUTH_URL, $CLIENT_ID, $CLIENT_SECRET;

    $expire_next = $_SESSION['expires_next'];

    if (!$_SESSION['token'] || !$expire_next || time() >= $expire_next - 2) {
        $client = new Client(["base_uri" => $AUTH_URL]);
        $response = $client->request('POST', 'token',
            ['form_params' => [
                'grant_type' => 'client_credentials',
                'client_id' => $CLIENT_ID,
                'client_secret' => $CLIENT_SECRET
            ]]
        );
        $body = $response->getBody();
        $data = json_decode($body->getContents());
        $_SESSION['token'] = $data->access_token;
        $_SESSION['expires_next']= time() + $data->expires_in;
    }

    return $_SESSION['token'];
}

function getMarketplaceDetails() {
    global $API_URL, $MARKETPLACE_UUID;

    $token = getAccessToken();
    $client = new Client(["base_uri" => $API_URL]);
    $response = $client->request('GET', "marketplaces/$MARKETPLACE_UUID",
        ['headers' => ['Authorization' => "Bearer $token"]]
    );
    $body = $response->getBody();
    return $body->getContents();
}

echo getMarketplaceDetails();
```

4.2. Basic

When you open your integration and/or production accounts, you will receive a login and password.

In this case, you only need to authenticate yourself in `Basic` mode.

For the Basic access, the usernames are *normally* the same in both environments.

5. DEFINING HEADERS

Headers	Description	Values to use
Accept	Determines the format of the contents that will be returned by the server.	Accept: application/json
Authorization	• **Basic**: Authentication token according to the "Basic Authentication over HTTPS" principle, i.e. 'Basic ' followed by the <login>:<password> string encoded in Base64. Conversion is automatically supported for clients such as Postman or Insomnia. • **OAuth2**: Access token according to the "Single Sign-on" principle, i.e. 'Bearer' followed by l'access_token. See above.	• **Basic example:** Authorization: Basic YWRtaW46YWRtaW4= • **Oauth2 example:** Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCIgOidUIiw...
Content-type	Determines the format of the contents sent to the server.	content-type: application/json
Method	See *our open-api* for the method to be used depending on the resource.	GET | POST | PUT | DELETE

Codes examples are provided in annex.

6. ENROLLING SELLERS

This section describes the *information that needs to be gathered*, as well as the four steps of the enrollment process:

Step	Operation	Status at the end of the stage
1	*Submission of detailed information about the seller*. This request generates a registration object and various nested objects (activity, board_member, etc.).	CREATED
2	You can then *edit your request* and *add the required KYC documents*.	AWAITING_SUBMISSION
3	Once the request is ready, *submit the request* to our compliance department *via* the API.	PENDING_VALIDATION
4	Finally, as soon as the request is validated, the corresponding seller is created, and you can *retrieve their login* and start the payments.	SUCCEEDED

To be automatically notified of the process, we recommend that you define webhooks for the registration object.

For more information on webhooks, see the *Marketplace API* Integration Guide.

Do not confuse the enrollment request and the merchant

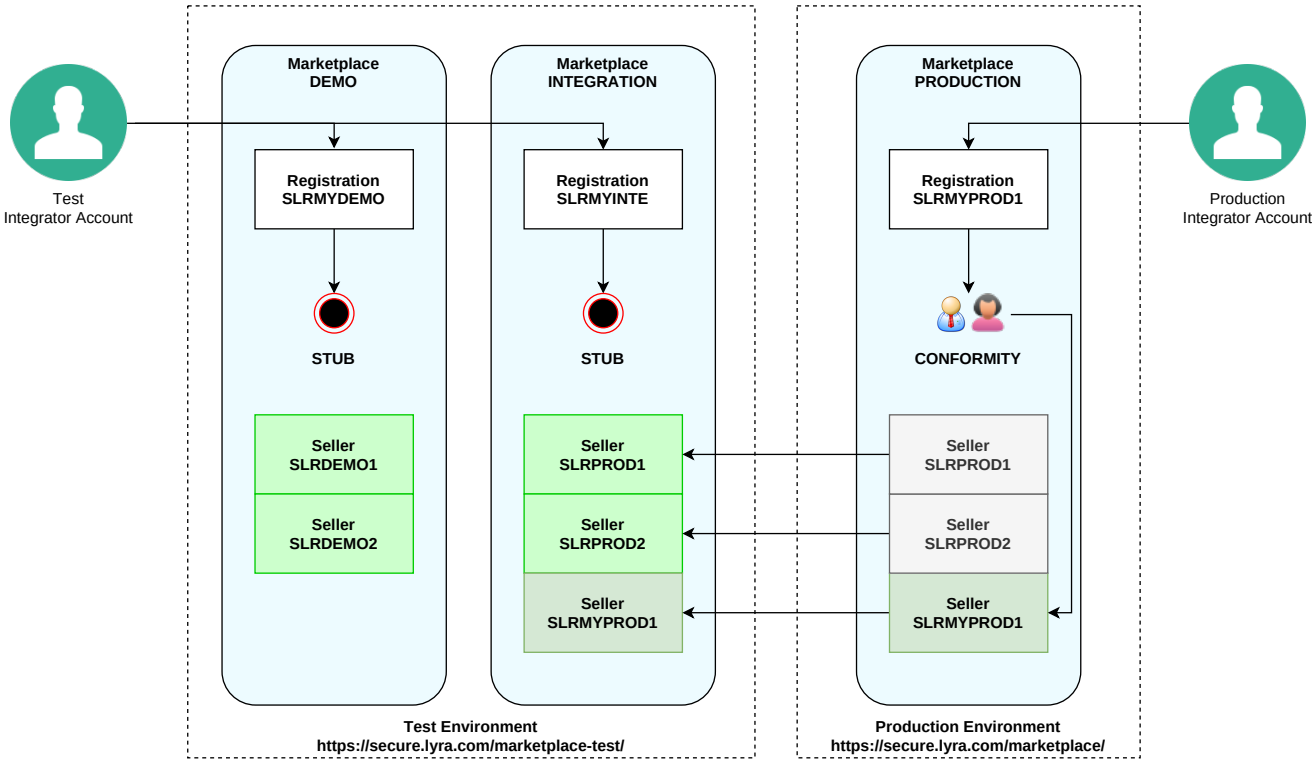
The enrollment request (registration) leads to the creation of a seller, but these are two different objects with separate uuids. Be careful not to confuse them.

6.1. Optimizing seller enrollment

As mentioned in the section covering the differences between the integration and production environments, your integration and production Marketplaces are separate from each other. There is one small exception: production sellers are shared with integration sellers.

This exception is intended to make your Integration Marketplace environment closer to real-world production conditions.

The diagram below is based on the premise that you have a demo, integration and production Marketplace. The following statements are provided for additional information. Note that the names of the sellers may be different, and the names of the registrations are for illustration purposes only.



Only enrollment requests (registration) from your production Marketplace are checked for compliance and result in the creation of the corresponding seller (see registration SLRMYPROD1 → seller SLRMYPROD1).

You can test your enrollment requests on your demo and integration Marketplace. However, they are capped and do not result in the creation of a seller (see registration SLRMYDEMO1 and SLRMYINTE1).

Two test sellers (seller) are provided in each environment (see SLRDEMO1, SLRDEMO2, SLRPROD1 and SLRPROD2 in the diagram above).

They allow you to perform payment tests in your integration environment as soon as you can access it.

> However, be very careful if you plan to make actual payments on the two test production vendors (SLRPRO1 and SLRPROD2), as they are associated with test disbursement accounts. You should then cancel the transactions before the capture (i.e. before midnight of the payment day). Otherwise, you would have to make a refund, which would incur additional transaction fees. To avoid any difficulties, we recommend that you do not make any payments to these sellers. If you want to test a real payment, we suggest you rather transfer it to the marketplace operator or to an existing sub-merchant.

Production vendors are automatically created in your integration Marketplace (see SLRPROD1, SLRPROD2 and SLRMYPROD1).

With this in mind, we recommend that you:

1. Carry out formal enrollment tests in your demo and integration Marketplaces (if available).
2. After that, *if possible*, perform a number of enrollments in production mode to confirm that your process is compliant with regulatory requirements.

6.2. Information required for the enrollment request (registration)

The enrollment request consists of a main `registration` object and various sub-objects.

6.2.1. Registration object details

Property	Description	Format / Permissible Values	Mandatory (X) or default value
marketplace	Marketplace identifier (`uuid`)	ans..36	X
reference	Your internal reference	ans..255	
external_ref	Your seller's technical reference. This reference, which must be unique if filled in, can be used instead of the seller's identifier when creating an order. This allows you to have an ID that can be used in the integration and production environment and to be able to store the vendor `uuid`. ℹ You can add an external reference to an existing enrollment request. However, you cannot update an already defined external reference.	ans.50	
description	Seller description.	ans..255	X
name	Legal name (company name as indicated on the KBIS).	ans..255	X
nature	Legal name (*idem* `name`).	ans..255	
legal_form	Legal form of the seller. See *Legal_form list*.	n4	
title	For natural entities: title	`MR, MS, MRS`	X if `person_type = PP`
first_name	For natural entities: First name	ans...63	X if `person_type = PP`
last_name	For natural entities: Last name	ans...63	X if `person_type = PP`
trading_name	For legal entities: Name of the brand under which the products are marketed.	ans..255	X if `person_type = PM`
vat_number	VAT number. When the seller is not subject to VAT, indicate the VAT number of the Marketplace.	ans..16	X
turnover	Estimated turnover (in cents).	Numeric >= 0	
cashout_delay	Disbursement delay (in days). The default value is 0 days, i.e. "as soon as possible after the capture". It may be useful to extend the delay to ensure that sufficient funds are available in case of a refund.	Numeric between 0 and 180.	0
iban	Payment IBAN.	ans..34	X if the `ibans` object is not populated and if the Marketplace uses only one currency.
ibans	Lists of seller's accounts (cf. `iban` object).	[cf. *Iban* `object`]	X if the Marketplace uses several currencies or if the `iban` is not specified.
address	Address of the seller's company.	[cf. *address*]	X
board	List of administrators.	[cf. *board_member*]	X
activities	List of activities of the seller. You must specify *only one* activity	[cf. *activity*].	X
vouchers	Lists of vouchers for which the seller is eligible	[cf. *voucher* object].	[]

6.2.2. Address details (address object)

Property	Description	Format / Permissible values	Mandatory (X) or default value
street_number	Street number	ans..5	
street	Street type	ans..255	
district	Address line 2	ans..127	
zipcode	Zip code	ans..64	
city	City	ans..128	
state	State/Province	ans..127	
country	2-letter country code (ISO 3166-1 alpha-2 code)	a2	X

6.2.3. Details of the administrator object (board_member object)

Property	Description	Format / Permissible Values	Mandatory (X) or default value
role	Status of the individual within the company: • LEGAL_REP: Legal representative • BENEFICIARY: beneficial owner	LEGAL_REP, BENEFICIARY	LEGAL_REP
title	Title of the individual	MR, MS, MRS	X
first_name	First name of the individual	ans..63	X
last_name	Last name of the individual	ans..63	X
birth_date	Birthdate of the individual	Date in YYYY-MM-DD format	X
birth_city	Place of birth of the individual	ans..128	X
nationality	Nationality of the individual (ISO 3166-1 alpha-2 code of the country)	a2	X
phone_number	Phone number of the individual	ans..32	
email	E-mail address of the individual	ans..150	
address	Personal address of the individual	cf. *address* object	X

6.2.4. Activity details (activity object)

Property	Description	Format / Permissible Values	Mandatory (X) or default value
name	Shop brand	Text	X
url	Website of the shop	ans..200	
registration_date	Date of registration of the activity	Date in YYYY-MM-DD format	
siret	SIRET identifier of the activity	ans..14	X if legal_identifier not filled
legal_identifier	Legal identifier of the activity	ans..20	X if siret not filled in.
mcc	MCC (Merchant Category Code) Example: 7997 for leisure clubs	n4	
naf	NAF code	an5	

6.2.5. Entering the IBANs

Each Marketplace currency must be associated with a disbursement account identified by its IBAN.

An account can receive several different currencies. On the other hand, it is not possible to enter the same account *several times*, be it for different currencies or for the same currency.

1) One default account

If your Marketplace supports only one currency (e.g. Euro), you can save the IBAN in an `iban` property of the POST query body. For example:

Extract from the POST /registrations request body

```
{ ... "iban":
 "FR1234567890000987654000", ... }
```

2) One account per currency

If your Marketplace supports multiple currencies, you *must* specify the recipient account(s) by currency. Fill in the list of IBANs by currency as follows:

Extract from the POST /registrations request body

```
{ ... "ibans": [ { "currencies":
 ["EUR", "GBP"], "iban":
 "FR1234567890000987654000" },
 { "currencies": ["CHF"], "iban":
 "CH12345675432199" } ], ... }
```

In this example, the transfers (cashout) of orders in Euro and Pounds sterling will be transfered to the account `FRFR1234567890000987654000`. If the orders have been made in Swiss francs, they will be sent to the account `CH12345675432199`.

Note that at the time of the enrollment request, the Marketplace must support the relevant currency. Otherwise, you will receive a *400 bad request error* `"Marketplace {marketplacename} does not support {currency} currency"`

Remember to add the documents!

For each IBAN defined this way, you must transmit the corresponding bank account details (see *Transmitting your KYC*).

6.2.6. Registering voucher contracts

In the case of Marketplaces that are eligible for certain acquirer contracts (i.e. CONECS and CVCONNECT), the contract reference must be recorded by each seller wishing to offer the corresponding payment method. The latter can therefore purpose the corresponding payment methods (Ticket-Restaurant or Chèque-Vacances) to the buyer during his items payment.

To do this, simply add a vouchers object to the enrollment request, representing a list of objects pairing a contract_number with a contract_type.

Example:

Extract from the POST /registrations request body

```
{ ... "vouchers": [ { "contract_number": "123456789", "contract_type": "CONECS" },
 { "contract_number": "234567890", "contract_type": "CVCONNECT" } ], ... }
```

Only one contract number can be registered per contract type.

How to identify the contract number?

Depending on the contract type, the contract_number can denote the merchant ID or the corresponding seller number. In all cases, this is the unique identifier linking the sub-merchant to the contract.

At CONECS, a VAD contract corresponds to a “VADS contract” (e-commerce).

The Conecs contract number to be provided corresponds to your technical identifier (IDCONECS) specific to e-commerce that generally begins with 1XXXX.

Conecs contract numbers specific to in-store sales generally begin with the letter C. *This type of contract number will not work for opening a contract on our platform*. It is therefore necessary to check the information transmitted by the sub-seller.

In case of a problem with a CONECS contract number, you can get in touch with CONECS support by e-mail **supportniveau2@conecs.fr** and provide the details of the sub-merchant, including their K-Bis number.

6.3. Creating a an enrollment request

Are all vendor details ready? Simply send them to us via the request:

POST **/items/** **registrations_create**

The transmitted elements are subject to an initial formal check before submission to the compliance department. Errors are indicated by a 400 return code and an error message.

Example of a minimal request body for the enrollment of a legal entity

```
POST https://secure.lyra.com/marketplace/v1/registrations/
```

```
{ "marketplace": "467ca516-da10-4744-9f8b-8f62f5965e2d", "person_type": "PM", "name": "Acme
 Corporation", "description": "Description du vendeur", "vat_number": "FR62881075493",
 "address": { "country": "FR" }, "board": [ { "role": "LEGAL_REP", "title": "MR", "first_name":
 "Thomas", "last_name": "Durand", "birth_date": "1949-09-17", "birth_city": "BONEVOIX",
 "nationality": "FR", "address": { "country": "FR" } } ], "activities": [ { "name": "Acme
 Tennis Ball", "siret": "99556699889944" } ], "ibans": [ { "currencies": ["EUR", "CHF"],
 "iban": "FR7630004000031234567890143" } ] }
```

Example of a minimal request body for the enrollment of a natural entity

```
{
 "marketplace": "467ca516-da10-4744-9f8b-8f62f5965e2d",
 "name": "BD EURL",
 "person_type": "PP",
 "title": "MR",
 "first_name": "Bastien",
 "last_name": "Durant",
 "description": "Bastien Durant's store",
    "vat_number": "FR62881075493",
 "address": {
  "country": "FR"
 },
 "board": [
  {
   "title": "MR",
   "email": "bd@eurl.com",
   "first_name": "Bastien",
   "last_name": "Durant",
   "birth_date": "1985-08-03",
   "birth_city": "FOIX",
   "nationality": "FR",
   "role": "LEGAL_REP",
   "address": {
    "country": "FR"
   }
  }
 ],
 "activities": [
  {
   "name": "BD Conseils",
   "siret": "88107549300018"
  }
 ],
    "ibans": [
  {
   "currencies": ["EUR"],
   "iban": "FR7630004000031234567890143"
  }
 ]
}
```


6.4. Modifying the enrollment request before submission

Using the unique uuid identifier of the enrollment request returned by the server during its creation, you can modify the enrollment request by making a call to the resource:

PUT **/registrations/{registration_uuid}** registrations_update

The request body to be used is the same as for creating an enrollment request... *except for the* board_member object.

Administrators have their own identifier. Therefore, if you want to *modify* an already registered administrator, you have to mention this uuid in the corresponding board_member object of the board list.

This uuid can be retrieved from the list of administrators returned by the server during the enrollment request, but you can also get it by calling:

GET **/registrations/{registration_uuid}** registrations_read

PUT /registrations/{registration_uuid} sample request body extract

```
{ ... "board": [ { "uuid": "b48cc6c4-6927-4e67-8743-1a97eb8450f8", "title": "MR",
 "first_name": "Walt E.", "last_name": "Coyote", "birth_date": "1974-01-13", "birth_city":
 "BURBANK", "nationality": "US", "role": "LEGAL_REP", "phone_number": "0531778899", "email":
 "willecoyote@wbs.com", "address": { "zipcode": "91501", "street_number": "12", "country":
 "US", "street": "Warner Bros Studios", "city": "BURBANK" } } ], ... }
```

However, if you do not want to change the list of administrators, send an empty board list. Otherwise, when you return the same administrator without the uuid, this will create a duplicate administrator.

Also note that it is not possible to delete an administrator.

At this point, the enrollment request has the CREATED status, and it will maintain this status until you submit the required KYC documents.

6.5. Transmitting your KYC (Know Your Customer) documents

This step will allow you to collect KYC documents in view of validating the seller.

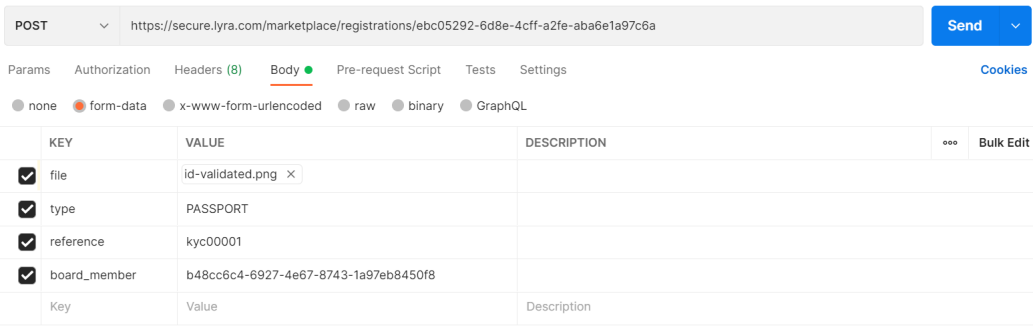
The transmission is done **file by file**, with a `multipart/form-data` request on the following resource:

POST **/registrations/{registration_uuid}/documents** **registrations_documents_create**

and with the following parameters:

Property	Description	Format / Permissible values	Mandatory (X)
file	The file to be transmitted	file (image or PDF)	X
type	Type of document: • `IDENTITY_CARD`: National identity card • `PASSPORT`: Passport • `RESIDENCE_PERMIT`: Residence permit • `IBAN`: IBAN • `KBIS`: KBIS extract • `COMPANY_BALANCE_SHEET`: Balance sheet • `JDD`: Proof of address	`IDENTITY_CARD`, `PASSPORT`, `RESIDENCE_PERMIT`, `IBAN`, `KBIS`, `COMPANY_BALANCE_SHEET`	X
reference	Technical reference of the document	ans..255	X
description	Description of the document	ans..255	
board_member	`uuid` of the administrator associated with the document.	an-36	X if `type` = `IDENTITY_CARD`, `PASSPORT`, `RESIDENCE_PERMIT`, `JDD`
additional_data	Allows you to record other information, such as the IBAN for the `IBAN` document type.		X for the `IBAN` file type and if the seller has several accounts.

Example of KYC POST with Postman



Example of a KYC POST response


```
{
    "uuid": "7e6c47bd-7a66-4aed-a3a8-56d1b3f815e6",
    "created_at": "2022-03-23T13:32:41.640641Z",
    "updated_at": "2022-03-23T13:32:46.592960Z",
    "registration": "ebc05292-6d8e-4cff-a2fe-aba6e1a97c6a",
    "board_member": "b48cc6c4-6927-8743-1a97eb8450f8",
    "type": "PASSPORT",
    "reference": "kyc00001",
    "additional_data": "",
    "status": "CREATED"
}
```


The 3 mandatory document types are:

- KBIS
- IBAN
- identity document, i.e. IDENTITY_CARD *or* PASSPORT *or* RESIDENCE_PERMIT

As soon as these have been successfully submitted, the enrollment request status will change to AWAITING_SUBMISSION.

6.6. Submitting the enrollment request

When the request is complete (mandatory KYC information and documents provided) and the request has taken the AWAITING_SUBMISSION status, you can call the resource:

POST **/registrations/{registration_uuid}/submit** **registrations_submit**

At the end of that call, the registration object takes the PENDING_VALIDATION status and no longer allows applying any changes.

6.7. Understanding and following up on the evolution of the enrollment request

At this stage, our sales and compliance department validates the transmitted information and documents. If necessary, they may contact you for clarification or additional information. You can then attach new documents to the request.

If you have defined webhooks for the registration object and the KYC, you will be notified of the processing stage of each document, and of the enrollment request.

If this is not the case, or if you are unsure whether a webhook has been received, you can still interrogate the corresponding resources and retrieve the latest status.

- For the enrollment request:

  GET **/registrations/{registration_uuid}** registrations_read

- For tracking the document statuses:

  GET **/registrations/{registration_uuid}/documents** registration_documents_list

If the information transmitted is valid, the documents will each successively take the VALIDATED status, then the status of the registration will change to SUCCEEDED.

However, it is possible that one of the transmitted *documents* is rejected (REJECTED) or marked as expired (EXPIRED). If it concerns one of the mandatory documents, the enrollment request status will return to CREATED. In this case, the request registration cycle must be repeated:

1. submission of a new corrective document
2. change of the enrollment request status to AWAITING_SUBMISSION
3. follow-up of the new submission

It is also possible that the *enrollment request* itself is rejected (FAILED). We recommend that you contact the sales department *via* our support team to find out why. The enrollment request status will revert to CREATED as soon as you update it, and you can resume the request submission process.

6.8. Retrieving the details of the generated seller

When the request results in the creation of a vendor, the registration object is assigned the status SUCCEEDED.

When the registrations_read resource is called, the seller property is then populated with the uuid of the seller associated with the request.

GET **/registrations/{registration_uuid}** registrations_read

Sample response body:

```
GET https://secure.lyra.com/marketplace-test/v1/registrations/ebc05292-6d8e-4cff-a2fe-aba6e1a97c6a
```

```
{ "uuid": "ebc05292-6d8e-4cff-a2fe-aba6e1a97c6a", "seller": "f8dcc611-bbaa-411a-8f28-ea2d6e4f49a8", "href": "http://secure.lyra.com/marketplace-test/v1/registrations/ebc05292-6d8e-4cff-a2fe-aba6e1a97c6a", "created_at": "2021-05-17T14:07:49.610694Z", "updated_at": "2021-05-31T13:08:26.148465Z", "marketplace": "2434c0a2-9d46-4e96-9553-1536c898625b", ... }
```

The seller's uuid then allows you to create orders on this seller.

This step is not necessary for creating orders if you have defined an external_ref for your enrollment request. You can then directly create the order using the value of external_ref with the seller_external_ref property.

If needed, you can view seller details by calling the sellers_read resource.

GET **/sellers/{seller_uuid}** registrations_read

Example of a request body for viewing seller details:

```
GET https://secure.lyra.com/marketplace-test/v1/sellers/a7dfdb01-bf23-4980-adc6-34c17ed3f887
```

```
{
  "uuid":"a7dfdb01-bf23-4980-adc6-34c17ed3f887",
  "external_ref":null,
  "href":"https://secure.lyra.com/marketplace-test/v1/sellers/a7dfdb01-bf23-4980-adc6-34c17ed3f887",
  "created_at":"2021-06-01T09:15:43.201011Z",
  "updated_at":"2022-03-28T14:45:49.522174Z",
  "marketplace":"57595c55-b096-41d8-9287-b98640de3f25",
  "reference":"SL271155",
  "description":"Everything for the kitchen",
  "bic":"",
  "iban":"",
  "status":"ACTIVE",
  "cashout_delay":4,
  "is_marketplace_seller":false,
  "links":{
    "items":{
      "href":"https://secure.lyra.com/marketplace-test/v1/sellers/a7dfdb01-bf23-4980-adc6-34c17ed3f887/items"
    },
    "transfers":{
      "href":"https://secure.lyra.com/marketplace-test/v1/sellers/a7dfdb01-bf23-4980-adc6-34c17ed3f887/transfers"
    }
  },
  "vouchers":[]
```



```
}
```

6.9. Understanding seller statuses

Sellers (seller) have a reduced number of statuses. Normally, you do not need to view or follow up on them. Here is a list of statuses and their description provided for information purposes:

Status	Meaning
INACTIVE	There are two possible cases: the seller has not been validated yet or has been disabled. In this case, they are not able to accept payments.
ACTIVE	The seller has been validated and can receive receipts and disbursements.
LOCKED	The seller was the subject of a modification request on your part, after their creation. They can receive receipts and, except in special cases (see chapter *Identifying disabled sellers* on page 23) continue to receive the corresponding disbursements.

The statutes of the seller objects may change without prior notice.

6.10. Updating an existing seller

After a seller has been created, you can edit their details by reusing the associated registration object.

Keep in mind that **any change results in a new validation cycle performed by the compliance department**.

Therefore, as soon as a registration status is modified to SUCCEEDED, it will return to the CREATED or AWAITING_SUBMISSION status depending on the presence and status of the mandatory KYCs, *without the need to add new ones (see below)*. You then have to resubmit your application for processing.

Some changes result in a temporary suspension of disbursements between the time when the change request is sent and the time the validation is effective. This is the case of a modification:

- Of the siret and/or legal_identifier fields of the activity object. In this case, attach the new KBIS.
- The identity of the legal representative. In this case, attach the ID of the new board_member.

Also note that at this time:

- The name field cannot be edited.
- It is also not possible to change an IBAN. In other words, the iban field cannot be edited. Moreover, if you have used the ibans field, you can delete an iban and reassign currencies to an existing iban, but you cannot modify or add one.

Statutes that do not allow modifications

It is not possible to update a registration with the following status:

- PENDING_VERIFICATION;
- SUSPENDED;
- CLOSED;
- REJECTED.

6.11. Identifying disabled sellers

Sometimes the seller is disabled, either temporarily or permanently. In this case, the corresponding registration will have the SUSPENDED or CLOSED status respectively.

The seller will take the INACTIVE status. It can no longer receive payments, and will no longer be subject to automatic disbursements.

7. APPENDIX

7.1. Leal_form list

Value list of the legal_form attribute :

Code	Label
1000	Individual entrepreneur
4110	National industrial or commercial public undertaking having a public accountant
4120	National industrial or commercial public undertaking not having a public accountant
4130	Public operator
4140	Local industrial or commercial public undertaking
4150	Management of an industrial or commercial local authority
5196	Cooperative savings and provident bank
5202	Commercial partnership
5203	Cooperative commercial partnership
5306	Ordinary limited partnership
5307	Ordinary cooperative limited partnership
5308	Partnership limited by shares
5309	Cooperative partnership limited by shares
5370	Professional holding company Partnership limited by shares
5385	Partnership limited by shares for a professional private practice
5410	National limited liability company
5415	Mixed economy limited liability company
5422	Limited liability real estate company for trade and industry
5426	Limited liability real estate management company
5430	Limited liability company for land development and rural establishment
5431	Limited liability company with mixed agricultural purposes
5432	Limited liability company for collective agricultural purposes
5442	Limited liability allocation company
5443	Cooperative construction limited liability company
5451	Cooperative consumer limited liability company
5453	Cooperative craftworkers' limited liability company
5454	Cooperative limited liability company for maritime purposes
5455	Cooperative limited liability transport company
5458	Cooperative workers' production limited liability company
5459	Limited liability union of cooperative companies
5460	Other cooperative limited liability company
5470	Professional holding company Limited liability company
5485	Limited liability company for a professional private practice
5498	Single-person limited liability company
5499	Limited liability company (with no further details)
5505	Workers' cooperative in the form of a public limited company with a board of directors
5510	National public limited company with a board of directors
5515	Mixed economy public limited company with a board of directors
5520	Open-end investment company with a board of directors
5522	Real estate public limited company for trade and industry with a board of directors
5525	Real estate investment public limited company with a board of directors
5530	Public limited company for land development and rural establishment with a board of directors
5531	Public limited company with mixed agricultural purposes with a board of directors

Code	Label
5532	Public limited company for collective agricultural purposes with a board of directors
5542	Public limited allocation company with a board of directors
5543	Public limited cooperative construction company with a board of directors
5546	Affordable housing public limited company with a board of directors
5547	Cooperative affordable housing production public limited company with a board of directors
5548	Real estate lending public limited company with a board of directors
5551	Consumer cooperative public limited company with a board of directors
5552	Retailers' cooperative public limited company with a board of directors
5553	Cooperative craftworkers' public limited company with a board of directors
5554	Cooperative public limited company for maritime purposes with a board of directors
5555	Cooperative transport public limited company with a board of directors
5558	Cooperative workers' production public limited company with a board of directors
5559	Union of cooperative companies public limited company with a board of directors
5560	Other cooperative public limited company with a board of directors
5570	Professional holding company Public liability company with a board of directors
5585	Public limited company for a professional private practice with a board of directors
5599	Public limited company with a board of directors (real estate public limited company)
5605	Workers' cooperative in the form of a public limited company with a management board
5610	National public limited company with a management board
5615	Mixed economy public limited company with a management board
5620	Fund in company form with a management board
5622	Real estate public limited company for trade and industry with a management board
5625	Real estate investment public limited company with a management board
5630	Public limited company for land development and rural establishment with a management board
5631	Public limited company with mixed agricultural purposes with a management board
5632	Public limited company for collective agricultural purposes with a management board
5642	Public limited allocation company with a management board
5643	Public limited cooperative construction company with a management board
5646	Affordable housing public limited company with a management board
5647	Cooperative affordable housing production public limited company with a management board
5648	Real estate lending public limited company with a management board
5651	Consumer cooperative public limited company with a management board
5652	Retailers' cooperative public limited company with a management board
5653	Cooperative craftworkers' public limited company with a management board
5654	Cooperative public limited company for maritime purposes with a management board
5655	Cooperative transport public limited company with a management board
5658	Cooperative workers' production public limited company with a management board
5659	Union of cooperative companies public limited company with a management board
5660	Other cooperative public limited company with a management board
5670	Professional holding company Public limited company with a management board
5685	Public limited company for a professional private practice with a management board
5699	Public limited company with a management board (real estate public limited company)
5710	Simplified joint stock company
5720	Simplified joint stock company with a single shareholder or single person simplified joint stock company
5770	Professional holding company Simplified joint stock company
5785	Simplified joint stock company for a professional private practice
5800	European company
6100	Savings and provident bank
6210	European Economic Interest Group (EEIG)
6220	Economic interest grouping
6316	Cooperative for shared use of agricultural equipment
6317	Agricultural Cooperative Organization

Code	Label
6318	Agricultural cooperative
6411	Mutual insurance company
6511	Inter-professional outpatient care companies
6521	Collective real estate investment civil-law company
6532	Civil-law company for collective agricultural purposes
6533	Collective farming grouping
6534	Agricultural land grouping
6535	Agricultural land grouping
6536	Forestry grouping
6537	Pastoral grouping
6538	Land and rural grouping
6539	Land civil-law company
6540	Property civil-law company
6541	Property civil-law company for construction and sale
6542	Civil-law allocation company
6543	Civil construction cooperative company
6544	Property civil-law company for gradual accession to ownership
6551	Civil-law cooperative consumer company
6554	Civil-law company for maritime purposes
6558	Civil-law cooperative company between doctors
6560	Other civil-law cooperative company
6561	Civil-law professional partnership of lawyers
6562	Civil-law professional partnership of legal counsel
6563	Civil-law professional partnership of solicitors
6564	Civil-law professional partnership of bailiffs
6565	Civil-law professional partnership of notaries
6566	Civil-law professional partnership of valuers and auctioneers
6567	Civil-law professional partnership of clerks of the commercial court
6568	Civil-law professional partnership of legal advisors
6569	Civil-law professional partnership of auditors
6571	Civil-law professional partnership of doctors
6572	Civil-law professional partnership of dentists
6573	Civil-law professional partnership of nurses
6574	Civil-law professional partnership of masseurs and physiotherapists
6575	Civil-law professional partnership of directors of medical analysis laboratories
6576	Civil-law professional partnership of veterinary surgeons
6577	Civil-law professional partnership of expert surveyors
6578	Civil-law professional partnership of architects
6585	Other civil-law professional partnership
6588	Dairy civil-law partnership
6589	Resourcing civil-law partnership
6595	Local mutual lending bank
6596	Mutual agricultural lending bank
6597	Civil-law partnership for agricultural operations
6598	Limited liability farm
6599	Other civil-law company
6901	Other private-law person registered in the trade and companies register
7111	Constitutional authority
7112	Independent administrative authority
7113	Ministry
7120	Central department of a ministry
7150	Ministry of Defense department

Code	Label
7160	Decentralized department of a ministry (excluding Defense) with national jurisdiction
7171	Decentralized State department with (inter-)regional jurisdiction
7172	Decentralized State department with (inter-)departmental jurisdiction
7179	(Other) decentralized State department with territorial jurisdiction
7190	National school not having legal entity status
7210	Municipality and new municipality
7220	Department
7225	Overseas collectivity and territory
7229	(Other) territorial collectivity
7230	Region
7312	Associated municipality and delegated municipality
7313	Section of a municipality
7314	Conurbation
7321	Authorized trade union association
7322	Urban land association
7323	Land consolidation association
7331	Local public information undertaking
7340	Metropolitan center
7341	Sector of a municipality
7342	Urban district
7343	Urban community
7344	Metropolis
7345	Inter-municipality syndicate with multiple aims
7346	Community of districts
7347	Community of towns
7348	Community of an urban area
7349	Other local non-specialist public cooperation entity or agreement
7351	Inter-departmental institution or agreement
7352	Inter-regional institution or agreement
7353	Inter-municipality syndicate with a single aim
7354	Mixed closed syndicate
7355	Mixed open syndicate
7356	Trade union committee for the management of shared property of municipalities
7357	Center for territorial and rural balance
7361	Municipal social welfare center
7362	Education funding authority
7363	Municipal lending bank
7364	Hospital establishment
7365	Inter-hospital syndicate
7366	Local public social and medical/social undertaking
7367	Inter-municipality social welfare center
7371	Public affordable housing office
7372	Departmental fire and safety department
7373	Local public cultural undertaking
7378	Management of a local authority of an administrative nature
7379	(Other) local public administrative undertaking
7381	Consular body
7382	National public undertaking having a central administration role
7383	National public undertaking of a scientific, cultural and professional nature
7384	Other national public teaching undertaking
7385	Other national public administrative undertaking with limited territorial jurisdiction
7389	National public undertaking of an administrative nature

Code	Label
7410	Public interest grouping
7430	Public religious undertaking of Alsace-Lorraine
7450	Public administrative undertaking, army assistance centers
7470	Publicly-managed cooperative healthcare group
7490	Other administrative law legal entity
8110	General social security scheme
8120	Special social security scheme
8130	Supplementary pension institution
8140	Agricultural social mutual fund
8150	Sickness scheme for non-employed, non-agricultural workers
8160	Retirement scheme not forming part of the general social security scheme
8170	Unemployment insurance scheme
8190	Other social welfare scheme
8210	Mutual fund
8250	Mutual agricultural insurance
8290	Other mutual organization
8310	Central works council
8311	Works committee
8410	Employees' trade union
8420	Employers' association
8450	Professional body or similar
8470	Technical industrial center or professional committee for economic development
8490	Other professional organization
8510	Welfare institution
8520	Supplementary pension institution
9110	Association of co-owners
9150	Voluntary landowners' association
9210	Undeclared association
9220	Declared association
9221	Declared association for integration through work
9222	Intermediary association
9223	Grouping of employers
9224	Association of lawyers with individual professional liability
9230	Declared recognized public interest association
9240	Religious order
9260	Local law association (Bas-Rhin, Haut-Rhin and Moselle)
9300	Foundation
9900	Other private-law legal entity
9970	Privately-managed cooperative healthcare grouping

7.2. Examples of codes for obtaining the authentication token

- **Basic mode**

cURL example

```
curl -X GET \
https://secure.lyra.com/marketplace-test/marketplaces/57595c55-b096-41d8-9287-b98640de3f25 \
-H 'Authorization: Basic YWRtaW46YWRtaW4='
-H 'Content-Type: application/json'
-H 'cache-control: no-cache'
```

Example of a complete request in Python

```
import requests

def get_marketplace_details():
    uuid = "57595c55-b096-41d8-9287-b98640de3f25"
    url = f"https://secure.lyra.com/marketplace-test/marketplaces/{uuid}"
    return requests.get(url, auth=(login, password),
                        headers={'content-type': 'application/json'})
```

Example of a request in .NET

```
var myURL = "https://secure.lyra.com/marketplace-test/orders?expand=items"
HttpWebRequest myHttpWebRequest = (HttpWebRequest)WebRequest.Create(myURL);
myHttpWebRequest.ContentType = "application/json";
myHttpWebRequest.Accept = "application/json";
myHttpWebRequest.Method = "post";
string authInfo = userName + ":" + userPassword;
authInfo = Convert.ToBase64String(Encoding.Default.GetBytes(authInfo));
myHttpWebRequest.Headers["Authorization"] = "Basic " + authInfo;
```

- **Oauth2 mode:**

cURL example

```
curl -X GET \
https://secure.lyra.com/marketplace-test/v1/marketplaces/57595c55-b096-41d8-9287-b98640de3f25 \
-H 'authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCIgOi2lkIiA6ICJocXFZcjNkSlYycH...' \
-H 'cache-control: no-cache'
```

Example of a complete request in Python

```
import requests

def get_marketplace_details():
    uuid = "57595c55-b096-41d8-9287-b98640de3f25"
    url = f"https://secure.lyra.com/marketplace-test/v1/marketplaces/{uuid}"
    access_token = get_access_token()
    return requests.get(url, headers={'authorization': f"Bearer {access_token}",
                        'content-type': 'application/json'})
```