



# COLLECTING SOLUTION

## Onboarding sellers

Document version 2.3

# Contents

<b>1. HISTORY OF THE DOCUMENT.....</b>	<b>3</b>
<b>2. PRESENTATION.....</b>	<b>4</b>
<b>3. CONSULT THE API ON OPEN API.....</b>	<b>5</b>
<b>4. AUTHENTICATE.....</b>	<b>6</b>
<b>5. DEFINING HEADERS.....</b>	<b>8</b>
<b>6. ENROLLING SELLERS.....</b>	<b>9</b>
6.1. Optimizing seller enrollment.....	9
6.2. Information required for the enrollment request (registration).....	11
6.2.1. Registration object details.....	11
6.2.2. Address details (address object).....	12
6.2.3. Details of the administrator object (board_member object).....	12
6.2.4. Activity details (activity object).....	14
6.2.5. Entering the IBANs.....	15
6.2.6. Registering voucher contracts.....	16
6.3. Creating a an enrollment request.....	17
6.4. Modifying the enrollment request before submission.....	18
6.5. Transmitting your KYC documents.....	19
6.6. Submitting the enrollment request.....	20
6.7. Understanding and following up on the evolution of the enrollment request.....	21
6.8. Simulating different use cases in a test environment.....	22
6.9. Retrieving the details of the generated seller.....	23
6.10. Understanding seller statuses.....	24
6.11. Updating an existing seller.....	25
6.12. Identifying disabled sellers.....	25
<b>7. APPENDIX.....</b>	<b>26</b>
7.1. legal_form list.....	26
7.2. Examples of codes for obtaining the authentication token.....	34

## 1. HISTORY OF THE DOCUMENT

Version	Author	Date	Comment
2.3	Lyra Collect	5/16/2024	Update of chapters: <ul style="list-style-type: none"><li>• Transmit your KYC documents</li><li>• Entering the IBANs</li></ul>
2.2	Lyra Collect	4/26/2024	Update of chapters: <ul style="list-style-type: none"><li>• Creating a an enrollment request</li><li>• Modifying the enrollment request before submission</li></ul>
2.1	Lyra Collect	1/8/2024	Update of chapters: <ul style="list-style-type: none"><li>• <i>Authenticate</i></li><li>• <i>Understanding seller statuses</i></li></ul>
2.0	Lyra Collect	6/7/2023	Updated chapters <ul style="list-style-type: none"><li>• <i>Transmitting your KYC (Know Your Customer) documents</i></li><li>• <i>Modifying the enrollment request before submission</i></li></ul>
1.9	Lyra Collect	5/12/2023	<ul style="list-style-type: none"><li>• Addition of the chapter <i>Simulating different use cases in a test environment</i></li><li>• Update of the <i>Understanding and following up the evolution of the enrollment request</i> chapter</li></ul>
1.8	Lyra Collect	3/16/2023	Updated chapters <ul style="list-style-type: none"><li>• <i>Consult the API on open API</i></li><li>• <i>Authenticate</i></li><li>• <i>Defining headers</i></li><li>• <i>Examples of authentication codes</i></li></ul>
1.7	Lyra Collect	4/13/2022	Document overhaul. Added Single Sign-On authentication mode.

This document and its contents are confidential. It is not legally binding. Any reproduction and / or distribution of all or part of this document or its content to a third party is strictly prohibited or subject to prior written authorization from Lyra Collect. All rights reserved.

## 2. PRESENTATION

---

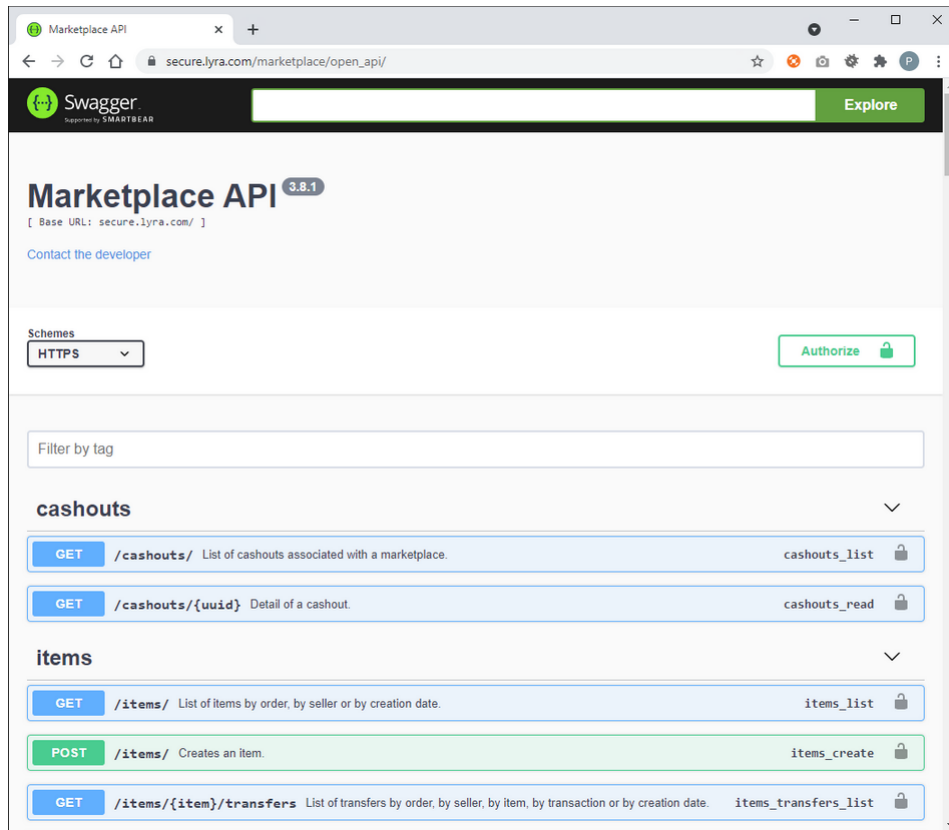
This document covers the operations that lead to [vendor enrollment](#) (resources, associated transactions, authentication, headers).

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

### 3. CONSULT THE API ON OPEN API

The Marketplace API uses [Swagger](#) for displaying the complete list of views at the following address:

[https://secure.lyra.com/marketplace/open\\_api/](https://secure.lyra.com/marketplace/open_api/)



Click on the views to access:

- a description of the action performed by the view;
- the expected parameters (and possibly the object) and their description;
- to any returned item;
- and status codes issued by the Marketplace API.

This document complements that resource. As it focuses on uses, it will regularly refer to each of the views, designating them by : method, path and name.

GET

/cashouts/

cashouts\_list

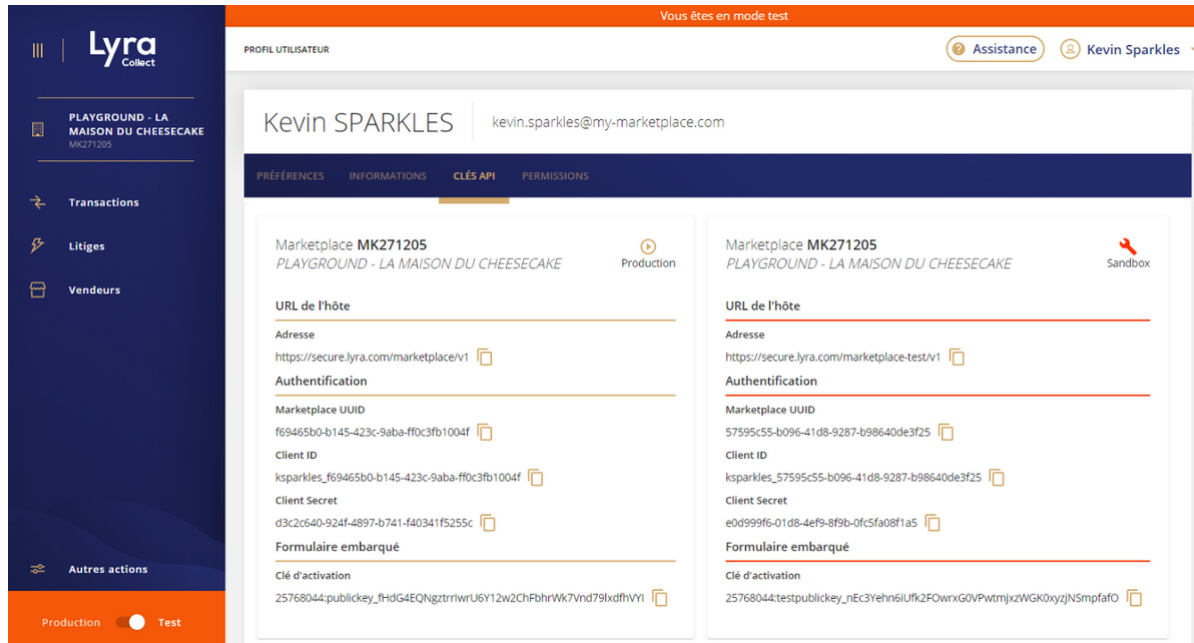


To simplify reading, only extracts of the items exchanged are sometimes included in this document. For your examples, you can use the [playground](#).

## 4. AUTHENTICATE

Authentication is in OAuth2 mode. You must have a technical account to access the Merchant Back Office.

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



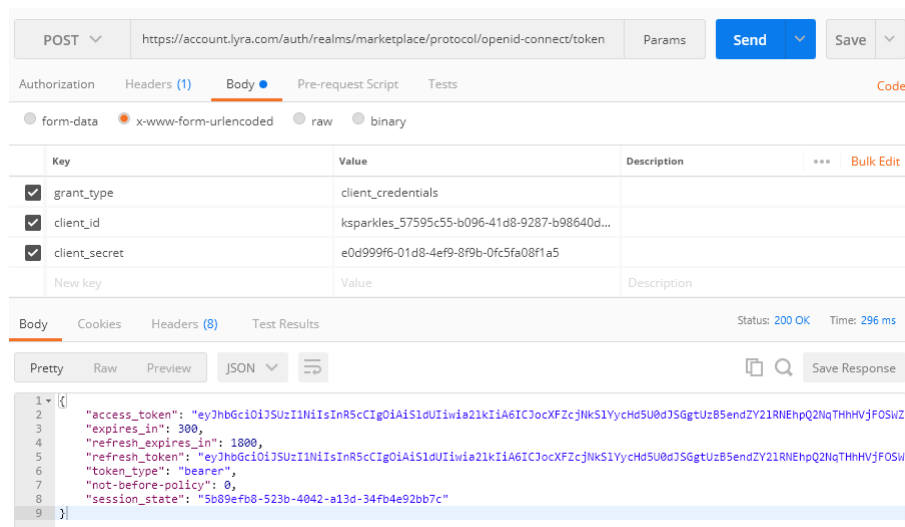
4. To generate an access token, make a request **POST** request to the following address:

<https://account.lyra.com/auth/realms/marketplace/protocol/openid-connect/token>

This is the same address for both environments.

5. Use the following parameters in format **application/x-www-form-urlencoded** :

- **grant\_type:client\_credentials**
- **client\_id:<CLIENT\_ID>**
- **client\_secret:<CLIENT\_SECRET>**



6. You can now create a query to retrieve the information from your test marketplace. To do so:

- Copy the authentication token (**access\_token**) into the **Authorization** header: **Bearer {access\_token}**

- Send your request to the environment address

[https://secure.lyra.com/marketplace-test/v1/marketplaces/{marketplace\\_uuid}](https://secure.lyra.com/marketplace-test/v1/marketplaces/{marketplace_uuid})

Example:

<https://secure.lyra.com/marketplace-test/v1/marketplaces/57595c55-b096-41d8-9287-b98640de3f25>



The authentication token has a lifetime of 5 minutes, and note that the refresh token is not used in the marketplace.

You must set up a method for retrieving a new token at no more than every 300 seconds (see PHP example below).

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

#### Example of obtaining the authentication token in PHP

```
<?php

require 'vendor/autoload.php';

use GuzzleHttp\Client;

$MARKETPLACE_UUID = "57595c55-b096-41d8-9287-b98640de3f25";
$USERNAME = 'ksparkles';
$CLIENT_ID = $USERNAME." ".$MARKETPLACE_UUID;
$CLIENT_SECRET = "e0d999f6-01d8-4ef9-8f9b-0fc5fa08f1a5";

$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();
```

## 5. DEFINING HEADERS

Headers	Description	Values to use
<b>Accept</b>	Determines the format of the contents that will be returned by the server. REST architecture that allows to perform data exchange in JSON format.	Accept: application/json
<b>Authorization</b>	<ul style="list-style-type: none"><li>• <b>OAuth2</b> : Access token according to the Single Sign-on principle, i.e. Bearer' followed by <code>access_token</code>. See above.</li><li>• <b>Mode Basic</b> : Identification token according to the "Basic Authentication over HTTPS principle, i.e. Basic' followed by the <code>&lt;login&gt;:&lt;password&gt;</code> string (login, password) encoded in Base64. Conversion is automatically supported for clients such as Postman or Insomnia.</li></ul>	<ul style="list-style-type: none"><li>• <b>OAuth2 example::</b> Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCIgOiUiw...</li><li>• <b>Basic example::</b> Authorization: Basic YWRtaW46YWRtaW4=</li></ul>
<b>Content-type</b>	Determines the format of the contents sent to the server.	content-type: application/json
<b>Method</b>	See <a href="#">our open-api</a> for the method to be used depending on the resource.	GET   POST   PUT   PATCH   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	<a href="#">Submission of detailed information about the seller</a> . This request generates a <a href="#">registration</a> object and various nested objects ( <a href="#">activity</a> , <a href="#">board_member</a> , etc.).	CREATED
2	You can then <a href="#">edit your request</a> and <a href="#">add the required KYC documents</a> .	AWAITING_SUBMISSION
3	Once the request is ready, <a href="#">submit the request</a> to our compliance department <i>via</i> the API.	PENDING_VERIFICATION
4	Finally, as soon as the request is validated, the corresponding <a href="#">seller</a> is created, and you can <a href="#">retrieve their login</a> and start the payments.	SUCCEEDED

To be automatically notified of the process, we recommend that you define webhooks for the object [registration](#).

For more information on webhooks, see the [Marketplace API](#) Integration Guide.



### Do not confuse the enrollment request and the merchant

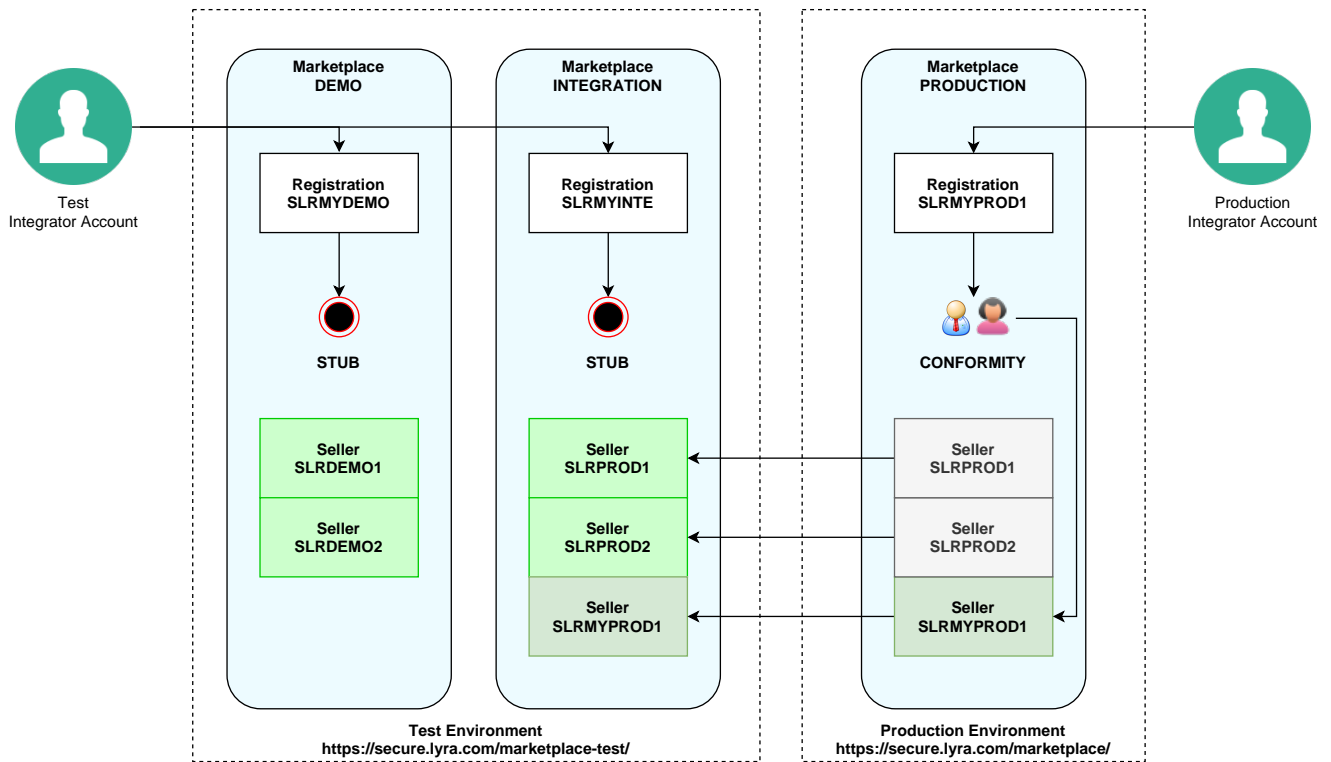
The enrollment request ([registration](#)) leads to the creation of a ([seller](#)), but these are two different objects with separate [uuids](#). Be careful not to confuse them.

### 6.1. Optimizing seller enrollment

Your integration and production Marketplaces are fully independent. 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. 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 Marketplaces. However, they are capped and do not result in the creation of a seller (see: registration SLRMYDEMO1 and SLRMYINTE1).

Two test (sellers) are provided with 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 using the two test production sellers (SLRPROD1 and SLRPROD2), as they are associated with test disbursement accounts.

In case of a real payment, you will have to cancel the transaction before the capture (i.e. before midnight on the day of the payment) or make a refund after the capture; this will involve additional transaction charges.

To avoid any difficulties, we recommend that you do not make any payments via 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 (if available) and integration Marketplaces;
2. After that, if possible, perform a number of real 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
<b>marketplace</b>	Marketplace identifier ( <b>uuid</b> )	ans..36	X
<b>reference</b>	Your internal reference	ans..255	
<b>external_ref</b>	<p>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 <b>uuid</b>.</p> <div> You can add an external reference to an existing enrollment request. However, you cannot update an already defined external reference.</div>	ans.50	
<b>description</b>	Seller description	ans..255	X
<b>name</b>	Legal name (company name as indicated on the KBIS).	ans..255	X
<b>nature</b>	Legal name ( <i>idem</i> <b>name</b> ).	ans..255	
<b>legal_form</b>	Legal form of the seller. See <a href="#">Legal_form</a> list.	n4	
<b>title</b>	For natural entities: title	<b>MR, MS, MRS</b>	X if <b>person_type</b> = PP
<b>first_name</b>	For natural entities: First name	ans...63	X if <b>person_type</b> = PP
<b>last_name</b>	For natural entities: Last name	ans...63	X if <b>person_type</b> = PP
<b>trading_name</b>	For legal entities: Name of the brand under which the products are marketed.	ans..255	X if <b>person_type</b> = PM
<b>vat_number</b>	VAT number. When the seller is not subject to VAT, indicate the VAT number of the Marketplace.	ans..16	X
<b>turnover</b>	Estimated turnover (in cents)	Numeric >= 0	
<b>cashout_delay</b>	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
<b>iban</b>	Payment IBAN	ans..34	X if the <b>ibans</b> object is not populated and if the

Property	Description	Format / Permissible values	Mandatory (X) or default value
			Marketplace uses only one currency.
<b>ibans</b>	Lists of seller's accounts (cf. <b>iban</b> object).	[cf. <b>iban</b> object]	X if the Marketplace uses several currencies or if the <b>iban</b> is not specified.
<b>address</b>	Address of the seller's company.	[cf. <b>address</b> ]	X
<b>board</b>	List of administrators.	[cf. <b>board_member</b> ]	X
<b>activities</b>	List of activities of the seller. You must specify <i>only one</i> activity	[cf. <b>activity</b> ].	X
<b>vouchers</b>	Lists of vouchers for which the seller is eligible	[cf. <b>voucher</b> object].	[]

### 6.2.2. Address details (address object)

Property	Description	Format / Permissible values	Mandatory (X) or default value
<b>street_number</b>	Street number	ans..5	
<b>street</b>	Street type	ans..255	
<b>district</b>	Address line 2	ans..127	
<b>zipcode</b>	Zip code	ans..64	
<b>city</b>	City	ans..128	
<b>state</b>	State/Province	ans..127	
<b>country</b>	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
<b>role</b>	Status of the individual within the company: <ul style="list-style-type: none"> <li><b>LEGAL_REP</b>: Legal representative</li> <li><b>BENEFICIARY</b>: beneficial owner</li> </ul>	<b>LEGAL_REP</b> , <b>BENEFICIARY</b>	<b>LEGAL_REP</b>
<b>title</b>	Title of the individual	MR, MS, MRS	X
<b>first_name</b>	First name of the individual	ans..63	X
<b>last_name</b>	Last name of the individual	ans..63	X
<b>birth_date</b>	Birthdate of the individual	Date in YYYY-MM-DD format	X
<b>birth_city</b>	Place of birth of the individual	ans..128	X

Property	Description	Format / Permissible Values	Mandatory (X) or default value
<b>nationality</b>	Nationality of the individual (ISO 3166-1 alpha-2 code of the country)	a2	X
<b>phone_number</b>	Phone number of the individual	ans..32	
<b>email</b>	E-mail address of the individual	ans..150	
<b>address</b>	Personal address of the individual	cf. <a href="#">address</a> 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 can 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 a property **iban** of the POST query body. For example:

#### Extract from the request body POST /registrations

```
{ ... "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 request body POST /registrations

```
{ ... "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 transferred 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** /registrations/

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/

```
{ "uuid": "354e6fd2-2568-4147-8b37-43acd6dc4881", "seller": null, "href": "https://secure.lyra.com/marketplace/registrations/354e6fd2-2568-4147-8b37-43acd6dc4881", "created_at": "2023-10-09T14:19:24.130103Z", "updated_at": "2023-10-09T14:19:30.181710Z", "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 shop", "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 **uuid** (unique identifier) of the enrollment request returned by the server during its creation, you can modify the **registration** by calling the resource:

**PATCH** `/registrations/{registration_uuid}`

registrations\_partial\_update

In this case, you can only send the desired modifications.

**PUT** `/registrations/{registration_uuid}`

registrations\_update

The request body to be used is the same as for creating a **registration**.



In both cases, make sure you specify the **uuid** of the **board\_member** that you would like to modify. Otherwise, you risk duplicating the **board member** in question.

Administrators have their own identifier, which can be retrieved from the list of administrators returned by the server when the enrollment request is made. You can also obtain it by calling:

**GET** `/registrations/{registration_uuid}`

registrations\_read

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.

### Sample extract of request body:

PUT `/registrations/{registration_uuid}`

```
{
  "marketplace": "d085b6bb-4b8e-4f23-b84a-c9e5fafaf3dd",
  "name": "Acme Corporation",
  "person_type": "PM",
  "description": "Seller description",
  "vat_number": "FRXX123456789",
  "address": {
    "country": "FR"
  },
  "board": [],
  "activities": [
    {
      "name": "Acme Tennis Ball",
      "siret": "15978525000012"
    }
  ],
  "ibans": [
    {
      "currencies": [
        "EUR",
        "CHF"
      ],
      "iban": "FR59441555432E4454227845122"
    }
  ]
}
```

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 documents

This step will allow you to collect KYC documents (= Know Your Customer) in view of validating the seller.

The transmission of files, also known as **uploading**, is carried out one by one, with a request of `multipart/form-data` type using the following resource:

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

`registrations_documents_create`

and with the following parameters:

Property	Description	Format / Permissible values	Mandatory (X)
<b>file</b>	The file to be transmitted	file (image or PDF)	X
<b>type</b>	Type of document: <ul style="list-style-type: none"><li>Identification and company articles of association (10 MB maximum per file):<ul style="list-style-type: none"><li><b>KBIS</b>: Extract from KBIS, <i>Certificate of incorporation</i> (mandatory)</li><li><b>BALANCE</b>: Balance sheet</li><li><b>STATUS</b>: Company status</li><li><b>COMPANY_OTHER</b>: Other documents related to the company</li></ul></li><li>Identity documents (4 MB maximum per file):<ul style="list-style-type: none"><li><b>IDENTITY_CARD</b>: National identity card</li><li><b>PASSPORT</b>: Passport</li><li><b>RESIDENCE_PERMIT</b>: Residence permit</li></ul></li><li>Account (10 MB maximum per file):<ul style="list-style-type: none"><li><b>IBAN</b></li></ul></li><li>Proof of address (10 MB maximum per file):<ul style="list-style-type: none"><li><b>ENERGY_PHONE_BILL</b>: Telephone bill</li><li><b>HOUSING_TAX</b>: Local taxes</li><li><b>PROOF_OF_RESIDENCY</b>: Proof of address</li><li><b>JDD</b>: Proof of address (same as for <b>PROOF_OF_RESIDENCY</b>)</li><li><b>ADDRESS_OTHER</b>: Other proof of address</li></ul></li></ul>	<b>IDENTITY_CARD</b> , <b>PASSPORT</b> , <b>RESIDENCE_PERMIT</b> , <b>KBIS</b> , <b>BALANCE</b> , <b>STATUS</b> , <b>COMPANY_OTHER</b> , <b>ENERGY_PHONE_BILL</b> , <b>HOUSING_TAX</b> , <b>PROOF_OF_RESIDENCY</b> , <b>JDD</b> , <b>ADDRESS_OTHER</b> , <b>IBAN</b>	X (au moins un type de document obligatoire par administrateur - board) :
<b>reference</b>	Technical reference of the document	ans..255	X
<b>description</b>	Description of the document	ans..255	

Property	Description	Format / Permissible values	Mandatory (X)
<b>board_member</b>	uuid of the administrator associated with the document.	an..36	X if it is one of the id or proof of address documents
<b>additional_data</b>	Allows you to record other information, such as comma-separated list of currencies for the IBAN document type. (e.g.: EUR,CHF,GBP).		X for the IBAN file type and if the seller has several accounts.

### Example of KYC POST with Postman

POST ▼ https://secure.lyra.com/marketplace/registrations/ebc05292-6d8e-4cff-a2fe-aba6e1a97c6a Send ▼

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/> file	id-validated.png <span>×</span>			
<input checked="" type="checkbox"/> type	PASSPORT			
<input checked="" type="checkbox"/> reference	kyc00001			
<input checked="" type="checkbox"/> board_member	b48cc6c4-6927-4e67-8743-1a97eb8450f8			
Key	Value	Description		

### 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"
}
```

In case of an exempted marketplace, the 3 types of mandatory documents are:

- The **KBIS** ;
- The **IBAN** ;
- as well as an identity document, i.e.: **IDENTITY\_CARD** or **PASSPORT** or **RESIDENCE\_PERMIT**.

If the marketplace is not exempt, you must add the company's articles of association (**STATUS**).

As soon as all the required documents 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 additional information.



If you have defined webhooks for the object `registration` 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 status `VALIDATED`, 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`).



### Reasons for rejection

If the document is set to `REJECTED` status, the `reject_reason` property guides you through the reasons for the rejection:

- `ILLEGIBLE`: ineligible document;
- `INCOMPLETE`: incomplete document;
- `INCONSISTENT`: irrelevant document;
- `OTHER`: other case.

If the rejection concerns one of the mandatory documents, the enrollment request status will return to `CREATED`. In this case, the request registration cycle must be restarted:

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

It is also possible that the enrollment request itself is:

- provisionally rejected (`FAILED`);

We recommend that you contact the sales department via our support team to find out why. the enrollment request status will return to `CREATED` as soon as you update it, and you can resume the request submission process.

- definitively rejected (`REJECTED`).

In this exceptional case, you can no longer can no longer modify the enrolment request in question.

## 6.8. Simulating different use cases in a test environment

---

The test environment enables you to force shifting an enrollment request and its KYC to the status of your choice, while respecting the overall process. For this, simply use the **description** field of these two objects (**registration** or **kyc**), to indicate the expected status. The object will take this status as soon as possible.

For example:

- You would like to test the expiration of a document.

In this case, mention "**EXPIRED**" (case insensitive) in the description of the document in question. The document will first take the **PENDING** status before taking the **EXPIRED** status upon the first synchronization.

- You would like to simulate an enrollment request rejection.

1. Create a **registration** containing "**REJECTED**" (case insensitive) in your description.

2. Specify the mandatory documents and submit the request.

The enrollment request will take the status of the first synchronization.

You can then add new documents, modify the enrollment request information - even change the description to define a new target status, which it will take as soon as possible, etc.



In the test environment, if the enrollment request status remains non-final for more than 7 days, it automatically changes to **CLOSED**.

## 6.9. 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": []
}
```

## 6.10. Understanding seller statuses

Sellers (seller) have a reduced number of statuses:

Status	Meaning
INACTIVE	Two possible cases: the seller has not been validated yet or has been disabled. They are not able to accept payments.
ACTIVE	The seller has been validated and can receive cash settlements and disbursements.
LOCKED	The seller was the subject of a modification request on your part, after its creation. They can receive receipts and, except in special cases (see chapter <a href="#">Identifying disabled sellers</a> on page 25) continue to receive the corresponding disbursements.



The status of the seller objects may change without notice.



### Pay attention to inactive sellers

The object seller is generated several minutes after the enrollment request (registration), but its status remains INACTIVE. Do not use the uuid of the seller in the registration for testing the seller's validity.



## 6.11. Updating an existing seller

---

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

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:

- fields `siret` and/or `legal_identifier` of the object `activity` ;

In this case, attach the new KBIS.

- the identity of the legal representative.

In this case, attach the ID of the new `board_member`.

Note also that in its current state:

- the field `name` cannot be modified;
- the field `iban` cannot be modified. Moreover, if you have used the field `ibans`, 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`.

## 6.12. 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. legal\_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

Code	Label
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
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

Code	Label
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
6318	Agricultural cooperative

Code	Label
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

Code	Label
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
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

Code	Label
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

Code	Label
7389	National public undertaking of an administrative nature
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



Code	Label
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 myHttpRequest = (HttpWebRequest)WebRequest.Create(myURL);
myHttpRequest.ContentType = "application/json";
myHttpRequest.Accept = "application/json";
myHttpRequest.Method = "post";
string authInfo = userName + ":" + userPassword;
authInfo = Convert.ToBase64String(Encoding.Default.GetBytes(authInfo));
myHttpRequest.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 eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXZWQjNkslYycH...' \
-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'})
```