



COLLECTING SOLUTION

REST Web service - Marketplace API

Implementation guide

Document version 1.3

Contents

1. HISTORY OF THE DOCUMENT.....	4
2. OBTAINING HELP.....	6
3. MARKETPLACE GLOSSARY.....	7
4. INTRODUCTION TO THE MARKETPLACE.....	8
Commission principle.....	8
5. PRÉSENTATION DU WEBSERVICE.....	9
5.1. Les grandes étapes du paiement.....	9
5.2. Web service resources.....	9
5.3. Prerequisites.....	10
6. UNDERSTANDING THE MARKETPLACE PAYMENT FLOW.....	11
7. IDENTIFYING YOURSELF DURING DATA EXCHANGE.....	14
8. UNDERSTANDING THE RETURN CODES OF THE HTTP STATUS SENT VIA WEB SERVICE.....	15
9. VIEWING THE SUB-MERCHANTS REGISTERED IN THE MARKETPLACE.....	16
10. MAKING A PAYMENT.....	17
10.1. Payment with capture delay.....	20
10.2. Creation of a token during a payment.....	20
10.3. Payment by token.....	21
10.4. Pre-authorized payments (manual validation).....	21
11. ANALYZING THE PAYMENT RESULT.....	24
12. TOKEN MANAGEMENT.....	26
12.1. Creating a token.....	26
12.2. Modifying an alias.....	26
12.3. Analyzing the result of a token request.....	27
12.4. Retrieving token details.....	27
13. PROCESSING THE RETURN TO THE MARKETPLACE.....	29
14. MODIFYING AN ORDER.....	30
15. CANCELING AN ORDER.....	32
16. REFUNDING A PAYMENT.....	33
16.1. Creating a refund request.....	33
16.2. Suivre l'évolution de la demande de remboursement.....	35
16.3. Modifying a refund request.....	36
16.4. Canceling a refund request.....	37
17. MANUAL TRIGGERING OF ITEM PAYMENT.....	38
18. VIEWING MARKETPLACE CASHOUTS.....	39
18.1. List of cashouts.....	39
18.2. Cashout details.....	40

19. DATA DICTIONARY.....	41
19.1. Address object.....	41
19.2. Alias object.....	41
19.3. Buyer object.....	42
19.4. Item object.....	42
19.5. Marketplace object.....	44
19.6. Order Object.....	45
19.7. Refund object.....	46
19.8. Seller object.....	47
19.9. Shipping object.....	48
19.10. Token object.....	49

1. HISTORY OF THE DOCUMENT

Version	Author	Date	Comment
1.3	Lyra Collect	16/07/2019	<p>Addition of the Cashouts resource. Update of the Items, Refunds and Order resources. Order object: addition of the <i>embedded_form_token</i> attribute. Addition of actions:</p> <ul style="list-style-type: none"> List the cashouts of a marketplace, Obtain cashout details, Edit a refund request, Cancel a refund request, Unlock the payment of a previously locked item. <p>Updated chapters:</p> <ul style="list-style-type: none"> Following up the request, Modifying an order, Presentation of the Web Services, Major stages of a payment, Understanding the marketplace payment flow, Making a payment. <p>Addition of chapters:</p> <ul style="list-style-type: none"> Manual triggering of item payment, Viewing marketplace cashouts, Modifying a refund request, Canceling a refund request, Pre-authorized payments (manual validation).
1.2	Lyra Collect	22/01/2019	<p>Addition of field definition for "order". Addition of the commission principle. Addition of Tokens and Refunds resources. Addition of Tokens and Refunds life cycle. Addition of the Cancel order action. Chapter Making a payment updated:</p> <ul style="list-style-type: none"> code samples updated, addition of the use of the Tokens resource. <p>Addition of chapters:</p> <ul style="list-style-type: none"> Analyzing the result of a token request. Retrieving token details Modifying an order Payment with capture delay Token object Alias object Refund object
1.1	Lyra Collect	12/07/2018	<p>Details added to chapters: addition of names within objects Removal of the transfers resource Addition of chapters:</p> <ul style="list-style-type: none"> Viewing the sub-merchants registered in the marketplace

Version	Author	Date	Comment
			<ul style="list-style-type: none">Refunding a payment
1.0	Lyra Collect	12/03/2015	Initial version

This document and its contents are confidential. It is not legally binding. No part of this document may be reproduced and/or forwarded in whole or in part to a third party without the prior written consent of Lyra Collect. All rights reserved.

2. OBTAINING HELP

Need some help? Please, check the FAQ on our website

<https://lyra.com/doc/fr/collect/faq/sitemap.html>

For technical inquiries or support, you can reach us from Monday to Friday, between 9 a.m. and 6 p.m.

by phone at:

0811900475

Service fee 0.06 € / min
+ call charge

by e-mail:

support-ecommerce@lyra-collect.com

and via your Expert Back Office, menu **Help > Contact support**

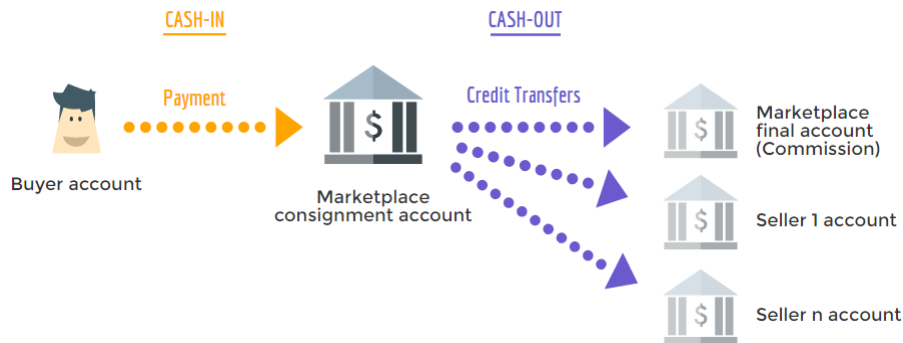
3. MARKETPLACE GLOSSARY

Vocabulary	Description
Cash-in	Payment made by the buyer to the Marketplace (CB, VISA, etc.).
Cash-out	Transfer made by the Marketplace to the Sellers.
Marketplace	A website offering items from different vendors/service providers.
Order	An order is composed of items, it corresponds to an order of the marketplace. Each item is associated with a seller.
Seller	A seller on the Marketplace, also called the provider. Can be a company (holder of a business registration number) or an individual.

4. INTRODUCTION TO THE MARKETPLACE

The payment platform offers payment services for Marketplaces that sell products or services on the Internet on behalf of sellers with whom they have signed a commercial agreement to this effect.

The mechanism of bank flows:



Vocabulary	Description
Cash-in	Payment made by the buyer to the Marketplace (CB, VISA, etc.).
Cash-out	Transfer made by the Marketplace to the Sellers.

Buyers' payments are distributed between sellers' payment accounts. The payment gateway then triggers credit transfers to the bank account of the Marketplace sellers.

Commission principle

An order contains a list of items, one for each item of the cart.

A special item is used to represent a commission taken by the marketplace from the seller's flow.

The seller of this item must be the seller marketplace, that is to say the one whose "**is_marketplace_seller**" field is "**true**".

Lyra Collect commissions are deducted on this item. The marketplace commissions must therefore cover the costs of Lyra Collect, for this, commission checks are carried out before the payment of the order.

An example of order creation is given in the chapter Making a payment on page 17.

5. PRÉSENTATION DU WEBSERVICE

Ce document présente le Webservice Marketplace qui permet de :

- Créer une commande
- Faire payer l'acheteur sur la page de paiement Lyra Collect ou *via* le formulaire embarqué;
- Consulter les informations d'une commande

Le Webservice Marketplace a été développé suivant le protocole REST.

5.1. Les grandes étapes du paiement

1. Si le site marchand a souscrit et installé le formulaire embarqué, l'acheteur saisit directement ses informations de paiement depuis le formulaire. Sinon, le **site marchand** redirige l'**acheteur** vers la page de paiement Lyra Collect.
La page de paiement affiche le montant total à régler.
2. Lyra Collect fait payer l'**acheteur** le montant total.
3. Si le paiement réussi, Lyra Collect crée les virements vers le(s) **Prestataire(s)** présent(s) dans le panier.
4. Lyra Collect effectue la remise en banque de la transaction effectuée par carte bancaire et envoie le fichier de virement à la **Banque**.

5.2. Web service resources

The Marketplace Web Services are available at the following address:

- **Test Mode** (integration phase): <https://secure.lyra.com/marketplace-test/>
- **Production**: <https://secure.lyra.com/marketplace/>

Resources of this API are reachable by different HTTP methods:

Resources	Action	HTTP method	URI
Marketplaces	Retrieve Marketplace details	GET	/marketplaces/{id_marketplace}
Sellers	Retrieve a provider	GET	/sellers/{id_seller}
	Retrieve the list of providers	GET	/marketplaces/{id_marketplace}
Orders	Create an order	POST	/orders?expand=items
	Modify an order	PUT	/orders/{id_order}
	Retrieve an order	GET	/orders/{id_order}
	Retrieve the list of Marketplace orders	GET	/orders
	Finalize an order and prepare the payment page	GET	/orders/{id_order}/execute
	Finalize an order and retrieve the token of the embedded form	GET	/orders/{id_order}/embedded-execute
	Cancel an order	DELETE	/orders/{id_order}
Items	Retrieve an element of an order	GET	/items/{id_item}
	Retrieve order details	GET	/orders/{id_order}/items

Resources	Action	HTTP method	URI
	Retrieve provider details	GET	/sellers/{id_seller}/items
	Create an element (item) inside an order	POST	/orders/{id_order}/items
	Unlock the payment of a previously locked item	POST	/items/{id_item}/activate
Refunds	Perform a refund	POST	/refunds
	Retrieve refund details	GET	/refunds/{id_refund}
	Retrieve the list of an order refunds	GET	/{id_order}/refunds
	Update a refund	PUT	/refunds/{id_refund}/
	Delete a refund	DELETE	/refunds/{id_refund}/
Tokens	Create/update a token	POST	/tokens
	Retrieve a simple token request	GET	/tokens/{id_token}
	Retrieve a token request linked to an order	GET	/tokens/{id_order}
Alias	Retrieve alias details	GET	/marketplaces/{id_marketplace}/alias/{id_alias}
Cashouts	List the cashouts for the sellers of a marketplace	GET	/cashouts
	Retrieve cashout details	GET	/cashouts/{id_cashout}

5.3. Prerequisites

Contact Lyra Collect to enable the access to the Marketplace and obtain the environment.

Prerequisites for the Marketplace

- Opt for the Marketplace offer.
After you select this offer, the payment platform will send you the data needed to access the Marketplace:
 - the **ID** of your Marketplace
 - a **login** and a **password** required for your identification.

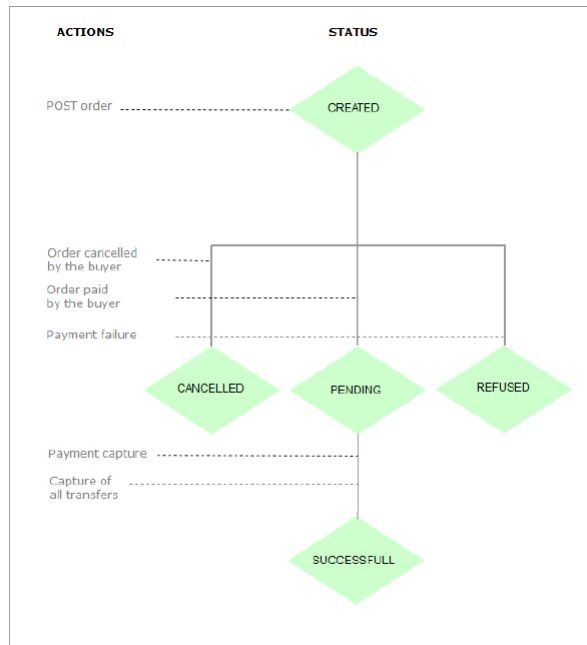
6. UNDERSTANDING THE MARKETPLACE PAYMENT FLOW

Below is the step-by-step process of a payment on the Marketplace:

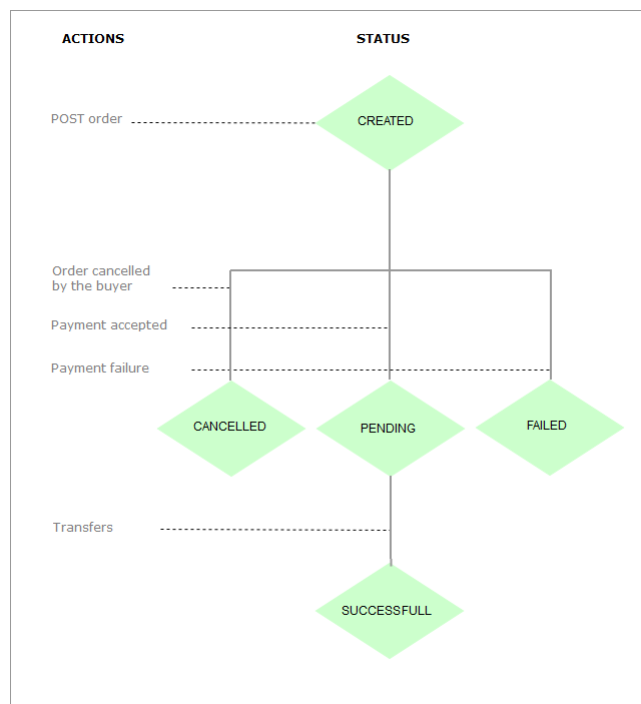
Stages	Actors	Actions
1	Buyer	Validates his/her cart on the Marketplace website.
2	Marketplace	Creates an Order via the REST API (POST ORDER) containing items.
3	Marketplace	Fixes the Order via the REST API (GET ORDER). Finalizes the Order and prepares the payment page or the token of the embedded form.
4	Payment platform	Creates the payment context of the Order. If the embedded form is enabled, returns a token (form-token) to the Marketplace, which must be inserted in the embedded form. Otherwise, returns a URL to the Marketplace for redirecting the buyer to the payment page.
5	Marketplace	If the embedded form is enabled, builds the form on the merchant website. Otherwise, redirects the buyer to the URL provided by the payment platform and retrieved by the GET ORDER.
6	Buyer	Fills in the bank details (+ 3D Secure authentication on the payment pages).
7	Payment platform	Validates and records the transaction.
8	Payment platform	Updates the status of the Items and the Order.
9	Payment platform	Sends a notification to the Marketplace (and, potentially, to the embedded form) indicating that the Order status has changed.
10	Marketplace	Calls the payment platform to find out the Order status and updates the system (GET ORDER).
11	Payment platform	Redirects the buyer to the Marketplace (return URL provided by the Marketplace = merchant website).
12	Payment platform	Performs a capture in the bank of the credit card (or another) transaction. Creates transfers in the Seller's shop. Sends the order file of the transfer to the bank.
13	Payment platform	Notifies the Marketplace about the modification of the Order and Items statuses.
14	Marketplace	Calls the payment platform to find out the Order status and updates the system (GET ORDER).
15	Marketplace	Manually validates the order file of the transfer in the its bank's interface.
16	Bank	Executes the requested transfers.

The status of resources varies throughout the whole process of the payment.

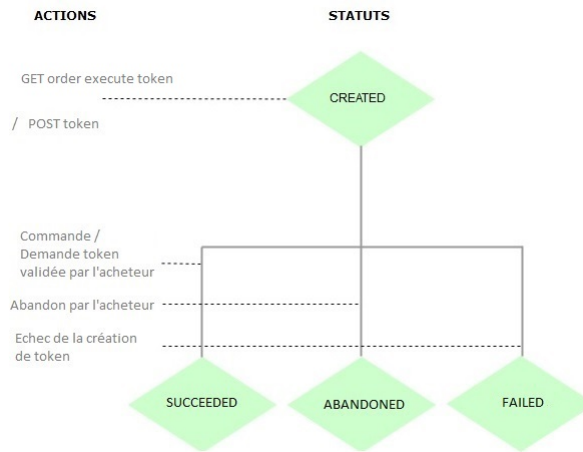
- **Order resource life cycle**



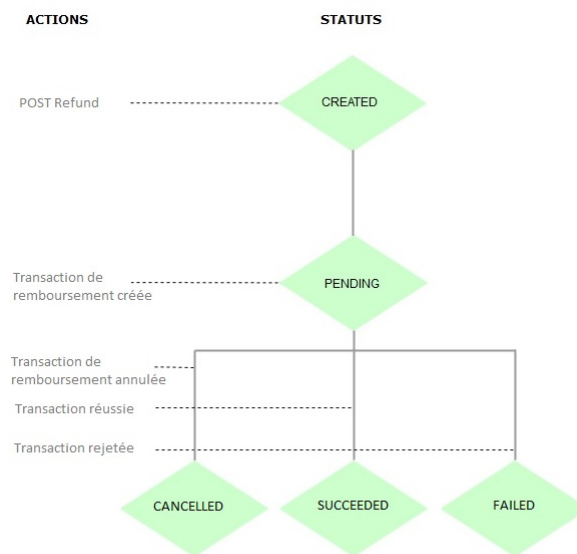
- **Item resource life cycle**



- **Tokens resource life cycle**



- **Refunds resource life cycle**



7. IDENTIFYING YOURSELF DURING DATA EXCHANGE

Identification is performed by means of an HTTP header.

The used method is **HTTP Basic Authentication**.

In each HTTP request, the header must contain the information allowing the Marketplace to authenticate itself when connecting to Marketplace Web Services.

Description of HTTP headers:

Headers	Description	Values to use
Accept	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'
Authorization: Description	Identification token according to the "Basic Authentication over HTTPS" principle. Consists of an identifier and a password of the API user separated by a ':', both encoded in Base64.	'Authorization: Basic YWRtaW46YWRtaW4='
Content-type	Determines the format of the contents sent to the server.	'content-type: application/json'
Method	See the table in the Web Service resources chapter to see the methods to use according to each resource.	GET POST PUT DELETE

The steps for building headers are:

1. Use the **Basic Authentication** method.
2. Specify the used method in the **Authorization** header: **Basic** followed by the login and the password (encoded in Base64) separated by a ':'.
3. Encode the obtained result in Base64.
4. Add the chain into "Basic".

Note:

*Do not forget to add a space after **Basic**.*

- Example **cURL**:

```
$ curl 'https://secure.lyra.com/marketplace-test/123456/orders/' -H
'Authorization: Basic YWRtaW46YWRtaW4=' -H
'Content-Type: application/json' -H 'Accept: application/json' --data
'{}'
-i
```

- Example of a complete request in **Python**:

```
r = requests.post(
    <target url>,
    data=<json data>,
    auth=(<api login>, <api password>),
    headers={'content-type': 'application/json'},
    verify=False
)
```

- Example of a request in **.NET**:

```
var myURL = "https://secure.lyra.com/marketplace-test/orders?expand=items"
HttpRequest myHttpRequest = (HttpRequest)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;
```

8. UNDERSTANDING THE RETURN CODES OF THE HTTP STATUS SENT VIA WEB SERVICE

Error code	Description
200 - 20X	Successfully processed request
401	Unauthorized access (problem in the Authorization header)
403	Forbidden access (the API user does not have the permission to perform this request)
400	Bad input data (example: the format of some parameters is not respected)
404	The requested object cannot be found (a non-existent UUID was requested)
500	Internal server error

Here are some examples of path analysis in order to help you quickly resolve the occurred issues:

Error 400:

- **"/?expand=items"** is absent at the end of the POST order request
- accents are not encoded in ANSI
- the item reference includes spaces.

9. VIEWING THE SUB-MERCHANTS REGISTERED IN THE MARKETPLACE

To view the sub-merchants registered in the marketplace, you must launch a call in GET mode.

```
GET https://secure.lyra.com/marketplaces/uuid/sellers
```

Example of a response to the GET call:

```
{
  "count": 1,
  "next": null,
  "previous": null,
  "results": [
    {
      "uuid": "eb94407d-c433-40ba-a5d3-d1baa0f7ed0a",
      "href": "https://secure.lyra.com/marketplace/sellers/eb94407d-c433-40ba-a5d3-d1baa0f7ed0a",
      "created_at": "2018-04-13T12:57:59.943702Z",
      "updated_at": "2018-04-13T12:57:59.943713Z",
      "marketplace": "30805a03-11ec-4447-93a5-243f39c89009",
      "reference": "MKPINTEG_Seller",
      "description": "",
      "bic": "",
      "iban": "",
      "status": "ACTIVE",
      "links": {
        "items": {
          "href": "https://secure.lyra.com/marketplace/sellers/eb94407d-c433-40ba-a5d3-d1baa0f7ed0a/items"
        },
        "transfers": {
          "href": "https://secure.lyra.com/marketplace/sellers/eb94407d-c433-40ba-a5d3-d1baa0f7ed0a/transfers"
        }
      }
    }
  ]
}
```


10. MAKING A PAYMENT

During a payment, unless the buyer has enabled and is using an embedded form, the buyer must be redirected to the payment page in order to finalize a purchase.

In order to make a payment, follow the steps below:

1. Create an order by using the POST method on the ORDER resource.
2. Fix the order to make the payment using the GET method:
 - either on the URL of the attribute: *execute* provided in the ORDER links,
 - or on the URL of the *execute-embedded* attribute in case of a payment by embedded form
3. In the GET response, retrieve:
 - either the redirection URL, attribute: *payment_url*,
 - or the *form_token* in case of a payment by embedded form.
4. Redirect the buyer to the URL of the payment page or display the embedded form using the *form_token*.

Step 1

To execute the order, the amount of the commission item must be greater than X% of the order total amount.

The X value is defined on our side for each marketplace and is the minimum percentage required for Lyra Collect to charge these fees.

These fees are deducted from the commission charged by the marketplace.

If the commission is insufficient, an error message is returned and the order is not executed.

For example, for a marketplace where the minimum charge is set at 5% the order below can be executed because the commission amount of the order is well above the required minimum.

Example of a POST ORDER call:

```
{
  "marketplace": "ebfb36ab-2d30-4326-adb9-e16b0c9a89f3",
  "reference": "MKP BURO example",
  "description": "office supplies order",
  "currency": "EUR",
  "url_return": "https://URLreturn.com",
  "items": [
    {
      "seller": "d0f80202-0676-4d8d-9247-f455f30aec1b",
      "reference": "commburo",
      "description": "commission BURO",
      "amount": 150,
      "is_commission": true
    },
    {
      "seller": "5d0ef88c-3345-4b33-948e-80e23d553b73",
      "reference": "STB1",
      "description": "blue pens",
      "amount": 1000,
      "is_commission": false
    }
  ],
  "buyer": {
    "type": "PRIVATE",
    "email": "jean.dupont@example.com",
    "phone_number": "33 (0)1 11 22 33 44",
    "reference": "ZhIsYM"
  },
  "shipping": {
    "shipping_method": "RELAY_POINT"
  }
}
```

```
}
```

Example of a POST ORDER response :

```
{
  "uuid": "1a2b51f8-9d62-408f-ab39-aa8e28ab15f0",
  "href": "https://secure.lyra.com/marketplace/orders/1a2b51f8-9d62-408f-ab39-aa8e28ab15f0",
  "created_at": "2019-01-21T16:46:46.517000Z",
  "updated_at": "2019-01-21T16:46:46.517000Z",
  "marketplace": "ebfb36ab-2d30-4326-adb9-e16b0c9a89f3",
  "reference": "MKP BURO example",
  "description": "Office supplies order",
  "alias": null,
  "buyer": [
    {
      "reference": "ZhIsYM",
      "title": null,
      "type": "PRIVATE",
      "first_name": null,
      "last_name": null,
      "legal_name": null,
      "phone_number": "33 (0)1 11 22 33 44",
      "email": "jean.dupont@acheteur.fr",
      "address": null
    }
  ],
  "shipping": [
    {
      "shipping_method": "RELAY_POINT",
      "delivery_company_name": null,
      "shipping_speed": null,
      "shipping_delay": null,
      "type": null,
      "first_name": null,
      "last_name": null,
      "legal_name": null,
      "phone_number": null,
      "address": null
    }
  ],
  "amount": null,
  "initial_amount": null,
  "currency": "EUR",
  "status": "CREATED",
  "webhook_result": null,
  "url_return": "https://URLreturn.com",
  "items": [
    {
      "uuid": "d33balab-793e-4d3d-943f-72b14a1e885b",
      "href": "https://secure.lyra.com/marketplace/items/d33balab-793e-4d3d-943f-72b14a1e885b",
      "created_at": "2019-01-21T16:46:46.510000Z",
      "updated_at": "2019-01-21T16:46:46.573000Z",
      "seller": "5d0ef88c-3345-4b33-948e-80e23d553b73",
      "order": "1a2b51f8-9d62-408f-ab39-aa8e28ab15f0",
      "reference": "STB1",
      "description": "blue pens",
      "type": "Manager",
      "amount": 1000,
      "quantity": 1,
      "transfer": null,
      "status": "CREATED",
      "links": null,
      "is_commission": false
    },
    {
      "uuid": "20034374-b425-4beb-91a8-a660ca5a3f9b",
      "href": "https://secure.lyra.com/marketplace/items/20034374-b425-4beb-91a8-a660ca5a3f9b",
      "created_at": "2019-01-21T16:46:46.502000Z",
      "updated_at": "2019-01-21T16:46:46.551000Z",
      "seller": "d0f80202-0676-4d8d-9247-f455f30aec1b",
      "order": "1a2b51f8-9d62-408f-ab39-aa8e28ab15f0",
      "reference": "commburo",
      "description": "commission BURO",
      "type": "Manager",
      "amount": 150,
      "quantity": 1,
      "transfer": null,
      "status": "CREATED",
      "links": null,
      "is_commission": true
    }
  ],
  "links": {
```

```

    "items": {
      "href": "https://secure.lyra.com/marketplace/orders/1a2b51f8-9d62-408f-ab39-aa8e28ab15f0/
items"
    },
    "refunds": {
      "href": "https://secure.lyra.com/marketplace/orders/1a2b51f8-9d62-408f-ab39-aa8e28ab15f0/
refunds"
    },
    "execute": {
      "href": "https://secure.lyra.com/marketplace/orders/1a2b51f8-9d62-408f-ab39-aa8e28ab15f0/
execute"
    },
    "execute-embedded": {
      "href": "https://secure.lyra.com/marketplace/orders/1a2b51f8-9d62-408f-ab39-aa8e28ab15f0/
execute-embedded"
    }
  },
  "vads_transaction_id": 0,
  "vads_transaction_date": null,
  "expected_capture_date": null,
  "capture_delay": null
}

```

Step 2

Example of a GET call via redirection to fix the order:

```

GET
https://secure.lyra.com/marketplaces/marketplace/orders/515abac9-6cb2-4e21-8a25-b08d7e41e43c/
execute

```

Example of a GET response:

```

{
  'payment_url': 'https://secure.lyra.com/vads-payment/
exec.refresh.a;jsessionid=HqKAcpvcgZQA29qCZDjXw4kS.marketplacevad01?
cacheId=450272311503195000050'
}

```

Example of a GET call via the embedded form to fix the order:

```

GET https://secure.lyra.com/marketplaces/marketplace/
orders/515abac9-6cb2-4e21-8a25b08d7e41e43c/embedded-execute

```

Example of a GET response:

```

{"form_token":"fa0yfV2FQuR3aak1SwsgcuZg195eyJhbW91bnQiOjEwOTAwLCJjdXJyZW5jeSI6IkVVUisiIm
1vZGUiOiJURVNUiwiidmVyc2lubiI6Mywib3JkZXJJZCI6IjVjNDk1N2E5LWNkOGEtNGJhZC1hMGNiLTgxMT
ZiMWE3ZDdmNiIsInNob3BOYW1lIjoiTllyYSBTTVMiLCJicmFuZFByaW9yaXR5IjpbIkJBTkNPT1RBQ1QiL
CJDQiIsIkUtQ0FSVEVCTEVVRSIsIk1BU1RFUkNBUkQiLCVGVQVVMVCJ9fX0a702"}

```

Note: the size of the form_token can be up to 8kb

10.1. Payment with capture delay

A time of capture of the transaction can be defined during the creation of an order. To do that, populate the "capture_delay" field with the desired number of days for the delay.

This value must be between 0 and 6:

```
POST https://secure.lyra.com/marketplace/orders
```

```
{
  "marketplace": "ebfb36ab-2d30-4326-adb9-e16b0c9a89f3",
  "reference": "MKP BURO example",
  "description": "Office supplies order",
  "currency": "EUR",
  "capture_delay": "3",
  ...
}
```

To change the capture date of the transaction associated with an already executed order (status PENDING), you cannot use the field "capture_delay". You must populate "**expected_capture_date**" with the desired capture date, between the current date D and D + 6. The format of the date is "YYYYMMDDhhmmss".

```
PUT https://secure.lyra.com/marketplace/orders/515abac9-6cb2-4e21-8a25-b08d7e41e43c
```

```
{
  "marketplace": "ebfb36ab-2d30-4326-adb9-e16b0c9a89f3",
  "reference": "MKP BURO example",
  "description": "Office supplies order",
  "currency": "EUR",
  "expected_capture_date": "20190425193000",
  ...
}
```

10.2. Creation of a token during a payment

Upon the creation of an order, it is possible to request the registration of the payment information in the form of a token that can be reused.

Use the "**execute/token**" command instead of "execute" during order validation.

Example of a GET call to fix the order:

```
GET https://secure.lyra.com/marketplaces/marketplace/orders/515abac9-6cb2-4e21-8a25-b08d7e41e43c/execute/token
```

10.3. Payment by token

To use a token, add the "alias" field to the creation of the order:

```
{
  "marketplace": "ebfb36ab-2d30-4326-adb9-e16b0c9a89f3",
  "reference": "MKP BURO example",
  "description": "Office supplies order",
  "currency": "EUR",
  "alias": "IDPAYMENT1",
  ...
}
```

Use the "**execute**" command on this order to create a payment with the token: "IDPAYMENT1".

Use the "**execute/token**" command to update the token with order data.

10.4. Pre-authorized payments (manual validation)

By default, Orders are validated automatically during step 2 of the payment (during GET <url>/execute).

It is also possible to dissociate the steps of payment authorization from those of a transaction capture.

This allows to make authorizations while the final transfer of cart items between the seller amount and the Marketplace fee amount is not yet known.

Step 1

In order to enable this feature, all you need to do is add the **awaiting_validation** field when creating the Order and assigning it the **true** value.

Example of a POST ORDER creation call in manual validation mode:

```
{
  "marketplace": "ebfb36ab-2d30-4326-adb9-e16b0c9a89f3",
  "reference": "MKP BURO example",
  "description": "Office supplies order",
  "currency": "EUR",
  "url_return": "https://URLreturn.com",
  "awaiting_validation" : true,
  "items": [
    {
      "seller": "5d0ef88c-3345-4b33-948e-80e23d553b73",
      "reference": "STB1",
      "description": "blue pens",
      "amount": 1150,
      "is_commission": false
    }
  ],
  "buyer":
  {
    "type": "PRIVATE",
    "email": "jean.dupont@buyer.fr",
    "phone_number": "33 (0)1 11 22 33 44",
    "reference": "ZhIsYM"
  },
  "shipping":
  {
    "shipping_method": "RELAY_POINT"
  }
}
```

In this example, the amount of the item (buyer amount) is known, but not the percentage of the sub-merchant and the operator fee (Marketplace).



Note: Please note that operator fees can already be specified in this step. If needed, they can be updated in step 4.

Step 2

Step 2 is executed in the same way as in case of standard payment and the redirection URL for proceeding to the payment is returned.

Example of a GET call to fix the order:

```
GET https://secure.lyra.com/marketplaces/marketplace/orders/7fac13b0-7ab9-4382-9073-11ddb38d4427/execute
```

Since the Order is waiting for validation, the verification steps of the minimum fee amount are not executed.

Step 3

Once the payment has been successfully made, the Order status changes to PENDING and the status of the corresponding transactions changes to **TO_VALIDATE**.

Step 4

This optional step allows to the amount of the operator fees if they were not provided in step 1.

It is also possible to add:

- a fee item
- and/or fees for items (see corresponding paragraph).

One or several items can also be deleted (standard function).

The total amount of the Order can be decreased but not increased.

On the other hand, the **awaiting_validation** field cannot be modified.

Step 5

The process is finalized with the validation of the Order. During this step one can make sure that the amount of the fee is sufficient.

In order to validate an Order, make a POST to the URL of the Order, attribute: *validate*.

Example of a POST ORDER call for manually validating it:

```
POST https://secure.lyra.com/marketplace/orders/7fac13b0-7ab9-4382-9073-11ddb38d4427/validate
```

Example of the validate POST ORDER response:

The Order details are returned. The *awaiting_validation* attribute of the Order then takes the **false** value and the status of the corresponding transactions changes from **TO_VALIDATE** to **PENDING**.

```
{
  "uuid": "7fac13b0-7ab9-4382-9073-11ddb38d4427",
  "href": "https://secure.lyra.com/marketplace/orders/7fac13b0-7ab9-4382-9073-11ddb38d4427",
  "created_at": "2019-05-27T09:30:24.434556Z",
  "updated_at": "2019-05-27T09:36:06.500936Z",
  "marketplace": "ebfb36ab-2d30-4326-adb9-e16b0c9a89f3",
  "reference": "MKP BURO 2 Example",
  "description": "Order of supplies",
  "alias": null,
  "awaiting_validation": false,
  "buyer": [
    {
      "reference": "ZhIsYM",
      "title": null,
      "type": "PRIVATE",
      "first_name": null,
      "last_name": null,
      "legal_name": null,
      "phone_number": "33 (0)1 11 22 33 44",
      "email": "jean.dupont@buyer.fr",
      "address": null
    }
  ],
  "shipping": [
    {
      "shipping_method": "RELAY_POINT",
      "delivery_company_name": null,
      "shipping_speed": null,
      "shipping_delay": null,
      "type": null,
      "first_name": null,
      "last_name": null,
      "legal_name": null,
      "phone_number": null,
      "address": {
        "street_number": "37",
        "street": "rue Marcel Philippe",
        "district": "Wallis and Futuna",
        "zipcode": "59259",
        "city": "Dupont",
        "state": "Meurthe-et-Moselle",
        "country": "FR"
      }
    }
  ],
  "payment_config": "SINGLE",
  "amount": 1150,
  "initial_amount": 1150,
  "currency": "EUR",
  "status": "PENDING",
  "webhook_result": null,
  "url_return": "https://URLreturn.com",
  "links": {
    "items": {
      "href": "https://secure.lyra.com/marketplace/orders/7fac13b0-7ab9-4382-9073-11ddb38d4427/items" },
    "transactions": {
      "href": "https://secure.lyra.com/marketplace/orders/7fac13b0-7ab9-4382-9073-11ddb38d4427/transactions" },
    "refunds": {
      "href": "https://secure.lyra.com/marketplace/orders/7fac13b0-7ab9-4382-9073-11ddb38d4427/refunds" }
    },
  "vads_transaction_id": 600001,
  "vads_transaction_date": "20190527093234",
  "expected_capture_date": "20190530093235",
  "capture_delay": 3
}
```



Important: The Order can be validated as long as the expiration date of the authorization request has not passed. If this date has passed, the payment takes the definitive **EXPIRED** status.

11. ANALYZING THE PAYMENT RESULT

To be able to process the payment result, Marketplace is notified via a URL provided to Lyra Collect, attribute: `webhook_url`.

A parameter provided in this notification will allow the Marketplace to execute a GET ORDER in order to retrieve the ORDER status and, therefore, know if the payment has been made successfully.

Example of a notification (webhook):

```
{'order': '515abac9-6cb2-4e21-8a25-b08d7e41e43c' }
```

Example of a GET ORDER call

```
GET https://secure.lyra.com/marketplace/orders/515abac9-6cb2-4e21-8a25-b08d7e41e43c/
```

Example of a GET ORDER response

```
{
  "uuid": "515abac9-6cb2-4e21-8a25-b08d7e41e43c",
  "href": "https://secure.lyra.com/marketplace/orders/515abac9-6cb2-4e21-8a25-b08d7e41e43c",
  "created_at": "2015-03-19T16:30:14.434Z",
  "updated_at": "2015-03-19T16:30:14.434Z",
  "marketplace": "9537e049-8862-400a-ae8d-da2ec9ca6051",
  "reference": "cmd00052",
  "description": "Order",
  "buyer": [{
    "reference": "nope775",
    "title": "M.",
    "type": "PRIVATE",
    "first_name": "Simone",
    "last_name": "Legendre",
    "phone_number": "02 13 06 95 27",
    "email": "ndurand@tiscali.fr",
    "address": {
      "street_number": "29",
      "street": "rue Besnard",
      "district": "Île-de-France",
      "zipcode": "83819",
      "city": "Roux",
      "state": "Hautes-Pyrénées",
      "country": "FR"
    }
  }],
  "shipping": [ {
    "delivery_company_name": "DHL",
    "address": {
      "street_number": "493",
      "street": "avenue Duhamel",
      "district": "Saint-Martin",
      "zipcode": "33980",
      "city": "Hamon",
      "state": "Charente-Maritime",
      "country": "FR"
    },
    "shipping_speed": "EXPRESS",
    "shipping_method": "ETICKET",
    "type": "PRIVATE",
    "first_name": "Luc",
    "last_name": "Leveque",
    "phone_number": "+33 (0)1 46 05 15 89"
  } ],
  "amount": null,
  "currency": "EUR",
  "status": "CREATED",
  "webhook_result": null,
  "url_return": "http://www.lyra-sms.com/",
  "links": {
    "items": {
      "href": "https://secure.lyra.com/marketplace/orders/515abac9-6cb2-4e21-8a25-b08d7e41e43c/items",
      "execute": {
        "href": "https://secure.lyra.com/marketplace/orders/515abac9-6cb2-4e21-8a25-b08d7e41e43c/execute"
      }
    }
  },
  "vads_transaction_id": 500003,
  "vads_transaction_date": ""
}
```


}

12. TOKEN MANAGEMENT

12.1. Creating a token

It is allowed to create a token with a request not linked to an order. To do so, call the TOKEN API as follow:

Example of a POST TOKEN call:

```
{
  "marketplace": "2434c0a2-9d46-4e96-9553-1536c898625b",
  "url_return": "http://www.my-website.com/",
  "buyer": {
    "type": "PRIVATE",
    "first_name": "Jean",
    "last_name": "Dupond",
    "email": "jean.dupond@lyra.fr",
    "phone_number": "1234",
    "reference": "Buyerr_1",
    "address": {
      "zipcode": "59259",
      "street_number": "37",
      "country": "FR",
      "street": "rue Marcel Philippe",
      "city": "Vignoux"
    }
  }
}
```

Example of a POST TOKEN response:

```
{
  "token": "51687172-cc9f-4ef2-8fe6-bbece95bd838",
  "payment_url": "https://secure.lyra.com/vads-payment/
exec.refresh.a;jsessionid=335D2aDb5eF8356Aed2cf3dF.vadpayment02inte01lbg?
cacheId=913355311811276000040"
}
```

12.2. Modifying an alias

To modify an existing token, fill the **alias_to_update** field in the POST TOKEN request with the token ID.

Example of a POST TOKEN call:

```
{
  "marketplace": "2434c0a2-9d46-4e96-9553-1536c898625b",
  "alias_to_update": "IDPAYMENT1",
  "url_return": "http://www.my-website.com/",
  "buyer": {
    "type": "PRIVATE",
    "first_name": "Jean",
    "last_name": "Dupond",
    "email": "jean.dupond@lyra.fr",
    "phone_number": "1234",
    "reference": "Buyer_1",
    "address": {
      "zipcode": "59259",
      "street_number": "37",
      "country": "FR",
      "street": "rue Marcel Philippe",
      "city": "Vignoux"
    }
  }
}
```

12.3. Analyzing the result of a token request

As for a payment, the Marketplace will be notified on the URL `webhook_url` of the evolution of a token request. The identifier thus transmitted will allow the Marketplace to make a GET TOKEN to recover the status of the token and therefore know if its creation succeeded.

During a payment with token creation, you will receive two notifications: one for the order and one for the token.

Example of a notification (webhook):

```
{"token": "d3329266-c8d6-421c-8d2a-10a8ffbcaef6"}
```

Example of a GET TOKEN call:

```
GET https://secure.lyra.com/marketplace/tokens/d3329266-c8d6-421c-8d2a-10a8ffbcaef6
```

Example of a GET TOKEN response:

```
{
  "uuid": "d3329266-c8d6-421c-8d2a-10a8ffbcaef6",
  "created_at": "2018-11-20T12:53:51.547541Z",
  "updated_at": "2018-11-20T12:53:52.513541Z",
  "marketplace": "9537e049-8862-400a-ae8d-da2ec9ca6051",
  "buyer": {
    "reference": "Buyer_1",
    "address": "MR",
    "type": "PRIVATE",
    "first_name": "Jean",
    "last_name": "Dupond",
    "phone_number": "012345678",
    "email": "jean.dupond@lyra.fr",
    "address": {
      "street_number": "37",
      "street": "rue Marcel Philippe",
      "district": null,
      "zipcode": "59259",
      "city": "Martin",
      "state": null,
      "country": "FR"
    }
  },
  "language": "fr",
  "url_return": "http://www.my-website.com/",
  "status": "SUCCEDED",
  "alias": "1144951ea7ab42989c97159b3dfc0382",
  "alias_to_update": null,
  "payment_url": "https://secure.lyra.com/vads-payment/exec.refresh.a;jsessionid=Cdb37C1CcfC5eA2BE82bDCA6?cacheId=913355311811206000040"
}
```

12.4. Retrieving token details

Once a token has been created, it is possible to obtain its information by calling the retail service of a token.

Example of a GET ALIAS call:

```
GET https://secure.lyra.com/marketplace/marketplaces/2434c0a2-9d46-4e96-9553-1536c898625bef6/alias/IDPAIEMENT1
```

Example of a GET ALIAS response:

```
{
  "brand": "CB",
  "expiry_month": "6",
  "expiry_year": "2023",
  "number": "497010XXXXXX0000",
}
```

```
"first_name": "Jean",  
"last_name": "Dupond",  
"email": "jean.dupond@lyra.fr",  
"creation_date": "2019-01-21T16:01:19Z"  
}
```

13. PROCESSING THE RETURN TO THE MARKETPLACE

In order to redirect the buyer to the Marketplace, the `url_return` field of the `ORDER` resource must be populated.

Note: the return to the Marketplace will only allow you to show visual context to the buyer. Do not use the received data for processing in the database or for checking the payment status.

14. MODIFYING AN ORDER

An order can be updated as long as its status is transient (**CREATED** or **PENDING**).

Therefore, the initial transaction must not be captured in the bank.

In case of an Order with the **CREATED** status, all values are editable, without limitations.

In case of an Order with the **PENDING** status, the PUT request is rejected if the two following values are identical to the ones in the previous record:

- **amount**, determined by the total amount of the items.

The update request cancels and replaces all items of the initial Order.

- **expected_capture_date** in UTC in the YYYYMMDDhhmmss format.

The comparison of expected_capture_date takes only the date into account.

For example, 20200101100000 (1 January 2020 at 10 a.m.) equals to 20200101180000 (1 January 2020 at 20 p.m.).

Furthermore, the amount cannot be higher than the initial amount of the order (i.e. value recorded in **initial_amount**).

Example of a PUT ORDER call:

PUT <https://secure.lyra.com/marketplace/orders/515abac9-6cb2-4e21-8a25-b08d7e41e43c>

Example of a PUT ORDER response

```
{
  "marketplace": "ebfb36ab-2d30-4326-adb9-e16b0c9a89f3",
  "reference": "MKP BURO example",
  "description": "Office supplies order",
  "currency": "EUR",
  "url_return": "https://URLreturn.com",
  "items": [
    {
      "seller": "d0f80202-0676-4d8d-9247-f455f30aec1b",
      "reference": "commburo",
      "description": "commission BURO",
      "amount": 150,
      "is_commission": true
    },
    {
      "seller": "5d0ef88c-3345-4b33-948e-80e23d553b73",
      "reference": "STV1",
      "description": "Green pens",
      "amount": 1000,
      "is_commission": false
    },
    {
      "seller": "d0f80202-0676-4d8d-9247-f455f30aec1b",
      "reference": "liv",
      "description": "Delivery",
      "amount": 500,
      "is_commission": false
    }
  ],
  "buyer": {
    "type": "PRIVATE",
    "email": "jean.dupont@buyer.fr",
    "phone_number": "+33 (0)1 11 22 33 44",
    "reference": "ZhIsYM"
  },
  "shipping": {
    "address": {
      "zipcode": "59123",
      "street_number": "37",
      "country": "FR",
      "street": "rue Marcel Philippe",
      "city": "Nantes"
    },
    "shipping_method": "RELAY_POINT"
  }
}
```

}

15. CANCELING AN ORDER

Orders can be cancelled before or after the payment since the payment has not been captured in the bank.

Example of a DELETE ORDER call:

```
DELETE https://secure.lyra.com/marketplace/orders/60c9dbf5-ff99-40fb-9fb6-a709005359f8
```


16. REFUNDING A PAYMENT

To refund a payment, you must indicate which order it is linked to and which seller will take on the cost of the refund.

To request a payment refund, follow the 2 following steps:

1. **Create a refund request**
2. **Follow up the request**

16.1. Creating a refund request

This step allows to create the refund request by using a POST request on the REFUND resource

POST: <https://secure.lyra.com/marketplace/refunds/>

When a refund request is created, the following elements are controlled by the API and are subject to a 400 returned error code if they are not verified:

- The refund request must have the PENDING or SUCCEEDED status.
- The seller that performs the refund must be one of the sellers of the refunded order.
- The refund must concern only one seller. If several sellers must refund a part of the same order, several separate refund requests must be created: one for each seller. The only exception is if the second seller of the request is the manager of the marketplace.

Note: *the refunded amount is not verified at this stage. The verification occurs later on, in scheduled process chains, that is reflected by the changes in the processing request status.*

Example of a POST REFUND call

In this example of the JSON content to post, a refund request of €130 is created, where €120 are refunded by the seller and €10 are refunded by the marketplace manager:

```
{
  "order": "9537e049-8862-400a-ae8d-da2ec9ca6051",
  "reference": "remb000045",
  "description": "Refund Mrs Smith 001",
  "currency": "EUR",
  "items": [
    {
      "seller": "dfc42a76-10b5-421a-91cd-c288c8265c92",
      "reference": "remb000045a",
      "description": "Phoneshop",
      "amount": 12000
    },
    {
      "seller": "975e2a43-7e72-438c-a2b2-b61347aa160c",
      "reference": "remb000045b",
      "description": "Manager",
      "amount": 1000
    }
  ]
}
```

Example of a POST REFUND response

The `amount` field in the response is calculated automatically based on the posted amounts of each of the parts.

```
{
  "uuid": "16ad9da8-b9cb-11e4-97c6-b1229586dec7",
  "href": "https://secure.lyra.com/marketplace/refunds/16ad9da8-b9cb-11e4-97c6-b1229586dec7",
  "created_at": "2018-06-08T12:36:56.681073Z",
  "updated_at": "2018-06-08T12:39:46.859402Z",
  "order": "9537e049-8862-400a-ae8d-da2ec9ca6051",
  "reference": "remb000045",
  "description": "Refund Mrs Smith 001",
  "amount": 13000,
  "currency": "EUR",
  "status": "CREATED",
  "items": [
    {
      "seller": "dfc42a76-10b5-421a-91cd-c288c8265c92",
      "reference": "remb000045a",
      "description": "Phoneshop",
      "amount": 12000,
      "is_commission": false
    },
    {
      "seller": "975e2a43-7e72-438c-a2b2-b61347aa160c",
      "reference": "remb000045b",
      "description": "Manager",
      "amount": 1000,
      "is_commission": false
    }
  ]
}
```

16.2. Suivre l'évolution de la demande de remboursement

Les remboursements sont exécutés en fonction des soldes disponibles du marchand et de la Marketplace. Par exemple, un remboursement de 100 euros avec 80 euros à la charge du marchand et 20 euros à la charge de la Marketplace ne sera exécuté que si :

- le solde disponible du sous-marchand est supérieur ou égal à 80 euros
- et si le solde disponible de la Marketplace est supérieur de 20 euros.

Pour suivre l'évolution de la demande, consultez le statut du REFUND par un GET:

GET : `https://secure.lyra.com/marketplace/refunds/{identifiant}/`

Une demande de remboursement peut passer par les états suivants au cours de son cycle de vie :

- **CREATED** : Demande de remboursement enregistrée par l'API, en attente de traitement
- **PENDING** : Demande adressée aux chaînes de traitement, en cours de traitement
- **SUCCEEDED** : Traitement du remboursement effectué avec succès
- **FAILED** : Demande de remboursement rejetée, par exemple en raison d'un solde insuffisant sur le compte du vendeur
- **CANCELLED** : Demande de remboursement annulée

Exemple réponse du GET REFUND

```
{
  "uuid": "16ad9da8-b9cb-11e4-97c6-b1229586dec7",
  "href": "https://secure.lyra.com/marketplace/refunds/16ad9da8-b9cb-11e4-97c6-b1229586dec7",
  "created_at": "2018-06-08T12:36:56.681073Z",
  "updated_at": "2018-06-08T12:51:21.241448Z",
  "order": "9537e049-8862-400a-ae8d-da2ec9ca6051",
  "reference": "remb000045",
  "description": "Remboursement Mme Lafont 001",
  "amount": 13000,
  "currency": "EUR",
  "status": "PENDING",
  "items":
  [
    {
      "seller": "dfc42a76-10b5-421a-91cd-c288c8265c92",
      "reference": "remb000045a",
      "description": "Phoneshop",
      "amount": 12000,
      "is_commission": false
    },
    {
      "seller": "975e2a43-7e72-438c-a2b2-b61347aa160c",
      "reference": "remb000045b",
      "description": "Gestionnaire",
      "amount": 1000
      "is_commission": false
    }
  ]
}
```

16.3. Modifying a refund request

A refund in progress can be modified within the limit of its initial amount as long as it has not been executed (CREATED or PENDING status).

The modification is made using the UUID of the corresponding refund, via the resource:

PUT: `https://secure.lyra.com/marketplace/refunds/{uuid}`



Note: If the refund amount is unchanged, the server returns an **HTTP 400** error.

Example of a PUT REFUND call

PUT `https://secure.lyra.com/marketplace/refunds/16ad9da8-b9cb-11e4-97c6-b1229586dec7`

```
{
  "order": "9537e049-8862-400a-ae8d-da2ec9ca6051",
  "reference": "remb000045",
  "description": "Refund Mrs Smith 001",
  "currency": "EUR",
  "items": [{
    "seller": "dfc42a76-10b5-421a-91cd-c288c8265c92",

    "reference": "remb000045a",
    "description": "Phoneshop",
    "amount": 9450,
    "is_commission": false
  },
  {
    "seller": "975e2a43-7e72-438c-a2b2-b61347aa160c",

    "reference": "remb000045b",
    "description": "Manager",
    "amount": 900,
    "is_commission": false
  }
  ]
}
```

Example of PUT REFUND response

```
{
  "uuid": "83402cc0-d969-443b-a72a-0f2fe9557879",
  "href": "https://secure.lyra.com/marketplace/refunds/83402cc0-d969-443b-a72a-0f2fe9557879",
  "created_at": "2019-06-11T09:31:59.659687Z",
  "updated_at": "2019-06-11T09:52:51.959406Z",
  "order": "67ca248e-99cd-4345-9795-5873bb6fd8f2",
  "reference": "remb000045",
  "description": "Refund Mrs Smith 001",
  "amount": 10905,
  "currency": "EUR",
  "status": "PENDING",
  "transaction": null,
  "items": [{
    "seller": "dfc42a76-10b5-421a-91cd-c288c8265c92",
    "refund": "83402cc0-d969-443b-a72a-0f2fe9557879",
    "item": "9f7c6d31-defb-4299-a3db-71bb214cae9e",
    "reference": "remb000045a",
    "description": "Phoneshop",
    "amount": 9450,
    "is_commission": false
  },
  {
    "seller": "975e2a43-7e72-438c-a2b2-b61347aa160c",
    "refund": "83402cc0-d969-443b-a72a-0f2fe9557879",
    "item": "7b99691b-0492-42ff-a819-302ac178fc7e",
    "reference": "remb000045b",
    "description": "Manager",
    "amount": 900,
    "is_commission": false
  }
  ]
}
```

16.4. Canceling a refund request

A refund in progress can be canceled as long as it has not been executed (CREATED or PENDING status).

DELETE: `https://secure.lyra.com/marketplace/refunds/{uuid}`

For example:

`https://secure.lyra.com/marketplace/refunds/83402cc0-d969-443b-a72a-0f2fe9557879`

In case of success, the server responds by a **HTTP 204** code (NO CONTENT).

If the refund status does not allow cancellations, the server returns an **HTTP 405** error.

17. MANUAL TRIGGERING OF ITEM PAYMENT

Normally, a payment for an item occurs automatically after the expiry of the delay for withholding the funds set by the seller related to the item.

The Marketplace may retain this payment in order to control when it is triggered.

For example, in case of a service that spreads over a period of time. In this case, the client has made his or her payment before the beginning of the service, but the seller will only receive the payment once the Marketplace verifies that the service has been successfully provided.

Then, the Marketplace:

- upon creation of the order, indicates the item(s) concerned by the withheld payment, setting **hold_payment** to **true**:

```
"items": [
  {
    "seller": "4d20a9d4-0526-4474-b452-e936dc25418d",
    "reference": "croisiere12345",
    "description": "Boat cruise",
    "amount": 245000,
    "quantity": 1,
    "is_commission": false,
    "hold_payment": true
  },
  {
    "seller": "4d20a9d4-0526-4474-b452-e936dc25418d",
    "reference": "nauticalchart",
    "description": "Nautical chart - Martinique",
    "amount": 1250, "quantity": 1,
    "is_commission": false
  },
  {
    "seller": "72ccc2ff-b455-4653-847e-deb6fee99f8d",
    "reference": "croisiere12345com",
    "description": "Boat cruise fee",
    "amount": 16450,
    "quantity": 1,
    "is_commission": true
  }
]
```

- and, when necessary, unlock the transfer by making a POST request (without a request body) with the following address:

```
POST /items/{id_item}/activate
```

In case the operation is successful, the request is redirected (status_code = 302) to the item details, which then indicates **hold_payment: false**.



Note:

- Payments can only be withheld for cart items of a Marketplace type seller (seller with "is_marketPlace = true").
- Once the item payment has been activated, it cannot be withheld again.
- The **hold_payment** field is not required. On the contrary, it is recommended to omit it (or to set it to "**hold_payment: null**") for all items whose transfer must be handled automatically, in order to distinguish the items that have been withheld from others.

18. VIEWING MARKETPLACE CASHOUTS

18.1. List of cashouts

The list of marketplace cashouts can be viewed at the following URL:

```
GET https://secure.lyra.com/marketplace/cashouts/
```

The results are returned as follows:

```
{
  "count": 3,
  "next": null,
  "previous": null,
  "results": [
    {
      "href": "/marketplace/cashouts/749e668e-526d-41ec-8e7c-e5afc3d89ddc",
      "uuid": "749e668e-526d-41ec-8e7c-e5afc3d89ddc",
      "seller": "b646ea68-a145-4a5a-8a6f-55e8f68643dd",
      "seller_external_ref": "ref039412",
      "ref": "demoeYDPnZuC",
      "status": "CREATED",
      "amount": 1900,
      "currency": "EUR",
      "captured_at": "2019-04-04",
      "capture_label": null
    },
    {
      "href": "/marketplace/cashouts/b92d904a-8613-4368-98b7-96d75a623d97",
      "uuid": "b92d904a-8613-4368-98b7-96d75a623d97",
      "seller": "f8dcc611-bbaa-411a-8f28-ea2d6e4f49a8",
      "seller_external_ref": "ref09523",
      "ref": "demoeYDPnZuC",
      "status": "CREATED",
      "amount": 1900,
      "currency": "EUR",
      "captured_at": "2019-03-03",
      "capture_label": null
    },
    {
      "href": "/marketplace/cashouts/f9132b0b-8d42-4409-b3e1-c1c6d711688b",
      "uuid": "f9132b0b-8d42-4409-b3e1-c1c6d711688b",
      "seller": "4d20a9d4-0526-4474-b452-e936dc25418d",
      "seller_external_ref": "ref012345",
      "ref": "demoeYDPnZuC",
      "status": "CREATED",
      "amount": 1450,
      "currency": "EUR",
      "captured_at": "2019-01-01",
      "capture_label": null
    }
  ]
}
```

These results are sorted by capture date in descending order ("captured_at" attribute).

They are displayed in a paginated manner. There are 100 cashouts per page.

The "next" and "previous" links, when they are populated, allow to navigate between pages.

It is also possible to filter the results of this list by their capture date.

In this case, the **capture_start_date** and/or **capture_end_date** parameters must be defined as follows:

```
GET https://.../marketplace/cashouts/?capture_start_date=2019-02-01&capture_end_date=2019-03-31
```

```
GET https://.../marketplace/cashouts/?capture_start_date=2019-02-01
```

```
GET https://.../marketplace/cashouts/?capture_end_date=2019-01-01
```

If the **capture_start_date** parameter is specified but **capture_end_date** is not, the latter is set to the current date (“today”).

On the other hand, if the **capture_start_date** parameter is omitted but **capture_end_date** is specified, the selection will take the cashouts from the beginning until the specified date.



Note: Cashouts with an unspecified “captured_at” attribute are excluded from the results.

18.2. Cashout details

The list of cashouts provides the URL for accessing the details of each of the cashouts, for example:

```
"/marketplace/cashouts/f9132b0b-8d42-4409-b3e1-c1c6d711688b".
```

A GET on the URL allows to retrieve information on the transfers and refund associated with the cashout (including order and item details):

```
GET https://.../marketplace/cashouts/f9132b0b-8d42-4409-b3e1-c1c6d711688b
```

```
{
  "href": "/marketplace/cashouts/f9132b0b-8d42-4409-b3e1-c1c6d711688b",
  "uuid": "f9132b0b-8d42-4409-b3e1-c1c6d711688b",
  "seller": "4d20a9d4-0526-4474-b452-e936dc25418d",
  "seller_external_ref": "ref012345",
  "ref": "demoeYDPnZuC",
  "status": "CREATED",
  "amount": 1450,
  "currency": "EUR",
  "captured_at": "2019-01-01",
  "capture_label": null,
  "transfers": [
    {
      "uuid": "0a5a8c7e-9c8b-4f16-8a29-0018c5aa20ef",
      "created_at": "2019-04-12T12:45:54.705798Z",
      "updated_at": "2019-05-06T12:15:09.942029Z",
      "item": {
        "uuid": "e8950426-f13c-4b18-8d27-17f2e6bbca8b",
        "ref": "demoeYDPnZuC",
        "desc": "Meal"
      }
    },
    "order": {
      "uuid": "0bfa5ebf-cd09-4fd8-bb78-465d17854b55",
      "ref": "TestMKP",
      "desc": " Test marketplace "
    },
    "amount": 1900,
    "currency": "EUR"
  ],
  "refunds": [
    {
      "uuid": "2760e1dd-252b-4109-816c-efbeafa2eaa2",
      "created_at": "2019-04-10T14:06:34.076796Z",
      "updated_at": "2019-04-10T14:06:35.719923Z",
      "order": {
        "uuid": "06f0fbd9-4a1f-4828-a014-aafda50df703",
        "ref": "TestMKP",
        "desc": "Test marketplace"
      },
      "item": {
        "uuid": "00f5a201-3d0c-49e7-9d05-0e706d04a385",
        "ref": "demoeYDPnZuC",
        "desc": "Meal"
      }
    },
    "amount": 450,
    "currency": "EUR"
  ]
}
```


19. DATA DICTIONARY

19.1. Address object

All the attributes of this object are required if at least one attribute is populated. Therefore, if no attributes are populated, no attributes are required.

Name	Description	Format	Mandatory	Example
street_number	Street number	an..5		12
street	Street name	ans..255		market street
district	Address supplement	ans..127		central district
zipcode	Zip code	an..64		75015
city	City	an.128		London
state	State	ans..127		London
country	Country	a2		FR for France GP for Guadeloupe PF for French Polynesia

19.2. Alias object

Name	Description	Format	Example
brand	Card brand	string	CB
expiry_month	Card expiry month between 1 and 12	n..2	3
expiry_year	Expiration year in 4 digits	n4	2023
number	Masked card number	string	497010XXXXXX0000
first_name	Buyer's first name	string	John
last_name	Buyer's last name	string	Dupond
email	Buyer's e-mail address	string	jean.dupond@example.com
creation_date	Token creation date	see example	2020-04-21T16:01:19Z

19.3. Buyer object

Name	Description	Format	Mandatory	Example
reference	buyer reference used in Marketplace	an..63	X	000012
legal_name	Company name	an..63		Dupond & Cie
title	buyer's civil status	an..63		Ms
type	type of buyer	enum	X	PRIVATE COMPANY
first_name	buyer's first name	an..63		Mary
last_name	buyer's last name	an..63		SMITH
phone_number	buyer's phone number	an..32	X	0612324565
email	Buyer's e-mail address	ans..150	X	m.smith@site.com
address	buyer's postal address	Address object		

19.4. Item object

Name	Description	Format	Mandatory	Example
uuid	Unique identifier of the service <i>Generated by Lyra Collect</i>	an..36	N/A	3ea5e574-e198-55d4-ba23-f9405ec4226f
href	URL to access this resource <i>Generated by Lyra Collect</i>	see example	N/A	https://secure.lyra.com/marketplace/items/3ea5e574-e198-55d4-ba23-f9405ec4226f
created_at	Date and time of creation of the resource <i>Generated by Lyra Collect</i>	see example	N/A	2015-01-17T09:39:54.948Z
updated_at	Date and time of the last modification of the resource <i>Generated by Lyra Collect</i>	see example	N/A	2015-01-17T09:39:54.948Z
seller	UUID of the provider	an..36	X	1ea5e574-e198-55d4-ba23-f9405ec4226a
order	UUID of the order	an..36	X	8ea5e574-e198-55d4-ba23-f9405ec4226c
reference	Reference of the item	an..32+"-"	X	0000000181
description	Description of the item	an..255		Item blablabla
amount	Amount of the item	n..12	X	50000
is_commission	indicates that this item is a commission	enum		true false false by default
status	Status of the item <i>Generated by Lyra Collect</i>	an..10	N/A	SUCCESSFUL
type	defines the type of the article in the cart	enum		see table below.

Type attribute values

Value	Description
FOOD_AND_GROCERY	Food and grocery
AUTOMOTIVE	Cars / Moto
ENTERTAINMENT	Entertainment / Culture
HOME_AND_GARDEN	Home and gardening
HOME_APPLIANCE	Household appliances
AUCTION_AND_GROUP_BUYING	Auctions and group purchasing
FLOWERS_AND_GIFTS	Flowers and presents
COMPUTER_AND_SOFTWARE	Computers and software
HEALTH_AND_BEAUTY	Health and beauty
SERVICE_FOR_INDIVIDUAL	Services for individuals
SERVICE_FOR_BUSINESS	Services for companies
SPORTS	Sports
CLOTHING_AND_ACCESSORIES	Clothes and accessories
TRAVEL	Travel
HOME_AUDIO_PHOTO_VIDEO	Sound, image and video
TELEPHONY	Telephony

19.5. Marketplace object

Name	Description	Format	Example
uuid	Unique identifier of the Marketplace <i>Generated by Lyra Collect</i>	an..36	6ea5e574-e198-55d4-ba23-f9405ec4226b
href	URL to access this resource <i>Generated by Lyra Collect</i>	see example	https://secure.lyra.com/marketplace/marketplaces/6ea5e574-e198-55d4-ba23-f9405ec4226b
created_at	Date and time of creation of the resource <i>Generated by Lyra Collect</i>	see example	2015-01-17T09:39:54.948Z
updated_at	Date and time of the last modification of the resource <i>Generated by Lyra Collect</i>	see example	2015-01-17T09:39:54.948Z
reference	Name of the Lyra Collect shop created for the Marketplace	an..255	Ex: Marketplace Shop name
description	Name of the Lyra Collect shop created for the Marketplace	an..255	Ex: Marketplace Shop name
bic	Bank code of the Marketplace bank account	an..11	CMCIC1234
iban	Account number of the Marketplace bank account	an..34	FR1212341234123412341234
vads_key	Identifier of the Lyra Collect shop created for the Marketplace <i>Generated by Lyra Collect</i>	an..8	12345678
vads_cert	Production certificate of the Lyra Collect shop created for the Marketplace <i>Generated by Lyra Collect</i>	an..16	123456789123456789
webhook_url	Notification URL of the Marketplace	an..1024	http://marketplace.com/page_traitement_notification.php
status	Status of the Marketplace resource <i>Generated by Lyra Collect</i>	an..10	ACTIVE
links	Links to the Sellers and Orders resources of the Marketplace <i>Generated by Lyra Collect</i>	see example	<pre>{ "sellers": { "href": "https://secure.lyra.com/marketplace/marketplaces/6ea5e574-e198-55d4-ba23-f9405ec4226b/sellers" }, "orders": { "href": "https://secure.lyra.com/marketplace/marketplaces/6ea5e574-e198-55d4-ba23-f9405ec4226b/orders" } }</pre>

19.6. Order Object

Name	Description	Format	Mandatory	Example
uuid	Unique identifier of the order <i>Generated by Lyra Collect</i>	an..36	N/A	8ea5e574-e198-55d4-ba23-f9405ec4226c
href	URL to access this resource <i>Generated by Lyra Collect</i>	see example	N/A	https://secure.lyra.com/marketplace/orders/8ea5e574-e198-55d4-ba23-f9405ec4226c
created_at	Date and time of creation of the resource <i>Generated by Lyra Collect</i>	see example	N/A	2015-01-17T09:39:54.948Z
updated_at	Date and time of the last modification of the resource <i>Generated by Lyra Collect</i>	see example	N/A	2015-01-17T09:39:54.948Z
marketplace	UUID of the Marketplace	an..36	X	6ea5e574-e198-55d4-ba23-f9405ec4226b
reference	Order reference	an..32+"-"	X	0000123
description	Order description	an..255		Order blablaba
vads_transaction_id	the transaction identifier Lyra Collect	n..6	N/A	379 <i>Note : Field depreciated at this place.</i>
vads_transaction_date	Date and time of the Lyra Collect transaction in the yyyyMMddHHmmss format	an..14	N/A	201503200945B <i>Note : Field depreciated at this place.</i>
buyer	Buyer details	Buyer object	X	
alias	Token ID in the payment gateway	ans..64		1144951ea7ab42989c97159b3dfc0382
shipping	Shipping details	Shipping object		
amount	Total amount of the order	n..12	N/A	50000
currency	Currency of the order (ISO 4217)	a3	X	EUR
language	Display language of the payment pages (ISO 639-1)	a2		FR
capture_delay	Indicates the delay (in days) before the capture	Positive integer between 0 and 6	No. Only available for creation (POST)	3
expected_capture_date	Expected capture date, in UTC timezone, in the YYYYMMDDhhmmss.	Date	No. Only available for update (PUT)	20180716083000
status	Order status <i>Generated by Lyra Collect</i>	an..10	N/A	PENDING
webhook_result	Result of the notification sent to the Marketplace <i>Generated by Lyra Collect</i>	n..3	N/A	200
url_return	Buyer's return URL on the Marketplace website	an..512		http://www.sitemarketplace.com/url_de_retour.html

Name	Description	Format	Mandatory	Example
links	Links to the Items resources <i>Generated by Lyra Collect</i>	see example	N/A	<pre>{ "items": { "href": "https://secure.lyra.com/ marketplace/orders/8ea5e574- e198-55d4- ba23-f9405ec4226c/items" } }</pre>
form_token	Token of the embedded form	an		<p>Example:</p> <pre>ffa0yfV2FQuR3aak1SwsgcuZg195eyJhbW91bnQi OjEwOTAwLCJjdXJyZW5jeSI6IkVVUilSm1vZG UiOiJURVNUliwidmVyc2lvbil6Mywib3JkZkZkZC I6IjVjNDk1N2E5LWNkOGEtNGJhZC1hMGNiLT gxMTZiMWE3ZDdmNiIsInNob3BOYW1lIjoiTHI yYSBTTVMiLCJicmFuZFByaW9yaXR5IjpbIkJB TkNPTIRBQ1QiLCJkQilskUtQ0FSVEVCTEVV RSIsIk1BU1RFUKNBukQjLCVGVQVVMVCj9fX 0a702</pre>

19.7. Refund object

Name	Description	Format	Mandatory
order	Identifier of the order to refund	an..36	X
reference	Technical reference of the refund	an..32	X
status	Status of refund	an..10	X
description	Description of the refund	ans..255	
currency	Currency of the order (ISO 4217) Only the EUR value is authorized	a3	X
items	List of refund parts to be made, by seller	enum	X
items.seller	Identifier of the seller who must take on this part of the refund	an..36	X
items.reference	Technical reference	an..32	X
items.description	Description	ans..255	
items.amount	Amount of the part to be refunded by the seller	n..12	X
items.item	Reference of the item being refunded within the original order	an..50	

19.8. Seller object

Name	Description	Format	Example
uuid	Unique item identifier of the provider <i>Generated by Lyra Collect</i>	an..36	<i>1ea5e574-e198-55d4-ba23-f9405ec4226a</i>
href	URL to access this resource <i>Generated by Lyra Collect</i>	see example	<i>https://secure.lyra.com/marketplace/sellers/1ea5e574-e198-55d4-ba23-f9405ec4226a</i>
created_at	Date and time of creation of the resource <i>Generated by Lyra Collect</i>	see example	<i>2015-01-17T09:39:54.948Z</i>
updated_at	Date and time of the last modification of the resource <i>Generated by Lyra Collect</i>	see example	<i>2015-01-17T09:39:54.948Z</i>
marketplace	UUID of the Marketplace	an..36	<i>6ea5e574-e198-55d4-ba23-f9405ec4226b</i>
reference	Identifier of the provider's Lyra Collect shop	an..255	<i>87654321</i>
description	Identifier of the provider's Lyra Collect shop	an..255	<i>Provider's shop</i>
bic	Bank code of the provider's bank account	an..11	<i>CMCIC1234</i>
iban	Account number of the provider's bank account	an..34	<i>FR1212341234123412341234</i>
is_marketplace_seller	allows to identify the marketplace seller. In the order, commission items (is_commission = true) must be linked to this seller. This seller is automatically created during the marketplace's creation.	boolean	<i>true false</i>
status	Status of the Marketplace resource <i>Generated by Lyra Collect</i>	an..10	<i>ACTIVE</i>
links	Links to the Items resources of the provider <i>Generated by Lyra Collect</i>	see example	<pre>{ "items": { "href": "https://secure.lyra.com/marketplace/sellers/1ea5e574-e198-55d4-ba23-f9405ec4226a/items" }, }</pre>

19.9. Shipping object

Name	Description	Format	Mandatory	Example
delivery_company_name	transporter's name	ans..128		
shipping_speed	shipping mode	enum		<i>STANDARD EXPRESS PRIORITY</i>
shipping_method	shipping method	enum	X	<i>RECLAIM_IN_SHOP RELAY_POINT RECLAIM_IN_STATION PACKAGE_DELIVERY_COMPANY ETICKET</i>
shipping_delay	speed related to delivery mode when shipping_speed = PRIORITY.	enum		<i>INFERIOR_EQUALS SUPERIOR IMMEDIATE ALWAYS</i>
type	type of recipient	enum		<i>PRIVATE COMPANY</i>
legal_name	company name	an..63		<i>Dupond & Cie</i>
name	recipient's name			
first_name	buyer's first name	an..63		<i>Mary</i>
last_name	buyer's last name	an..63		<i>SMITH</i>
phone_number	buyer's phone number	an..32		<i>0612324565</i>
address	buyer's postal address	Address object		

19.10. Token object

Name	Description	Format	Mandatory	Example
uuid	Unique identifier of the order <i>Generated by Lyra Collect</i>	an..36	N/A	<i>c730def9-e171-4f4f-92de-b7f09ed74b8d</i>
created_at	Date and time of creation of the resource <i>Generated by Lyra Collect</i>	see example	N/A	<i>2018-11-27T13:53:48.461191Z</i>
updated_at	Date and time of the last modification of the resource <i>Generated by Lyra Collect</i>	see example	N/A	<i>2018-11-27T13:53:48.461191Z</i>
marketplace	UUID of the Marketplace	an..36	N/A	<i>2434c0a2-9d46-4e96-9553-1536c898625b</i>
buyer	Buyer details	Buyer object	X	
language	Display language of the payment pages (ISO 639-1)	a2		<i>FR</i>
url_return	Buyer's return URL on the Marketplace website	an..512		<i>http://www.my-website.com</i>
status	Request status <i>Generated by Lyra Collect</i>	an..10	N/A	<i>CREATED</i>
alias	Token ID in the payment gateway	ans..64		<i>1144951ea7ab42989c97159b3dfc0382</i>
alias_to_update	Token ID to update	ans..64		<i>1144951ea7ab42989c97159b3dfc0382</i>
payment_url	URL to give to the buyer for the creation of the token	ans..255		