



**SOLUTION COLLECT**

# **API Webservices REST Marketplace**

## **Guide d'implémentation**

Version du document 2.4.2

# Sommaire

<b>1. HISTORIQUE DU DOCUMENT.....</b>	<b>4</b>
<b>2. GLOSSAIRE DE LA MARKETPLACE.....</b>	<b>7</b>
<b>3. PRÉSENTATION DE LA MARKETPLACE.....</b>	<b>8</b>
Principe de commission.....	8
Calcul de la commission minimale.....	8
Méthodes d'envoi.....	9
<b>4. PRÉSENTATION DU WEBSERVICE.....</b>	<b>11</b>
4.1. Les grandes étapes du paiement.....	11
4.2. Les Ressources du Webservice.....	11
4.3. Prérequis.....	12
<b>5. COMPRENDRE LE DÉROULEMENT D'UN PAIEMENT MARKETPLACE.....</b>	<b>13</b>
<b>6. S'IDENTIFIER LORS DES ÉCHANGES.....</b>	<b>16</b>
<b>7. COMPRENDRE LES INFORMATIONS DE LA MARKETPLACE.....</b>	<b>17</b>
<b>8. COMPRENDRE LES CODES RETOUR DE STATUT HTTP ENVOYÉS PAR LE WEBSERVICE.....</b>	<b>19</b>
<b>9. CONSULTER LES SOUS-MARCHANDS ENREGISTRÉS DANS LA MARKETPLACE.....</b>	<b>20</b>
<b>10. UTILISER DES WEBHOOKS.....</b>	<b>21</b>
10.1. Définir l'adresse du webhook.....	21
10.2. Accéder aux webhooks définis et accessibles sur votre marketplace.....	22
10.3. Enregistrer, modifier ou supprimer un webhook.....	23
10.3.1. Ajouter un webhook.....	23
10.3.2. Modifier un webhook.....	24
10.3.3. Supprimer un webhook.....	24
<b>11. RÉALISER UN PAIEMENT.....</b>	<b>25</b>
11.1. Paiement en N fois.....	29
11.1.1. Paiements réguliers : MULTI.....	30
11.1.2. Échéancier personnalisé : MULTI_EXT.....	31
11.2. Paiement avec délai de capture.....	32
11.3. Paiement avec création d'un alias.....	33
11.4. Paiement par alias.....	34
11.5. Paiement à l'initiative du marchand.....	35
11.6. Paiement en validation manuelle.....	36
11.7. Paiement par vouchers.....	40
11.7.1. Prérequis.....	40
11.7.2. Création et modification de la commande.....	41
11.7.3. Sélection des contrats à l'exécution.....	42
11.7.4. Après le paiement.....	43
11.7.5. Modification et annulation.....	44
11.8. Paiement à l'aide d'un lien persistant.....	45
11.8.1. Générer le lien.....	45
11.8.2. Adapter le délai d'expiration.....	45
<b>12. ANALYSER LE RÉSULTAT DU PAIEMENT.....</b>	<b>46</b>
<b>13. GESTION DES ALIAS.....</b>	<b>48</b>

13.1. Créer un token.....	48
13.2. Analyser le résultat d'une demande de token.....	50
13.3. Modifier un alias.....	51
13.4. Récupérer le détail d'un alias.....	53
<b>14. TRAITER LE RETOUR VERS LA MARKETPLACE.....</b>	<b>54</b>
<b>15. MODIFIER UNE COMMANDE.....</b>	<b>55</b>
15.1. Mise à jour d'une commande réglée en plusieurs fois.....	56
<b>16. ANNULER UNE COMMANDE.....</b>	<b>58</b>
<b>17. REMBOURSER UN PAIEMENT.....</b>	<b>59</b>
17.1. Créer la demande de remboursement.....	59
17.2. Suivre l'évolution de la demande de remboursement.....	61
17.3. Modifier une demande de remboursement.....	62
17.4. Annuler une demande de remboursement.....	63
<b>18. DÉCLENCHEMENT MANUEL DU PAIEMENT DES ITEMS.....</b>	<b>64</b>
<b>19. CONSULTER LES CASHOUTS DE LA MARKETPLACE.....</b>	<b>65</b>
19.1. Comprendre les cashouts.....	65
19.2. Cinématique du cashout.....	65
19.3. Identifier le cashout et les commandes associées.....	66
19.4. Liste des cashouts.....	66
19.5. Détails des cashouts.....	68
<b>20. GÉNÉRER UN CLIENT PAR OPENAPI.....</b>	<b>69</b>
<b>21. DICTIONNAIRE DE DONNÉES.....</b>	<b>70</b>
21.1. Objet Address.....	70
21.2. Objet Alias.....	70
21.3. Objet Buyer.....	71
21.4. Objet Item.....	71
21.5. Objet Marketplace.....	73
21.6. Objet Order.....	74
21.7. Objet Order voucher.....	75
21.8. Objet Refund.....	75
21.9. Objet Seller.....	77
21.10. Objet Shipping.....	78
21.11. Objet Token.....	79
21.12. Objet Transaction.....	80
<b>22. OBTENIR DE L'AIDE.....</b>	<b>84</b>

# 1. HISTORIQUE DU DOCUMENT

Version	Auteur	Date	Commentaire
2.4.2	Lyra Collect	25/11/2021	Mise à jour du chapitre <i>Accéder aux webhooks définis et accessibles sur votre marketplace</i> : correction de la requête exemple
2.4.1	Lyra Collect	17/11/2021	Mise à jour du chapitre <i>Objet Shipping</i> : correction de la valeur <code>PACKAGE_DELIVERY_COMPANY</code>
2.4	Lyra Collect	6/10/2021	<ul style="list-style-type: none"><li>Ajout du chapitre <i>Utiliser des webhooks</i>.</li><li>Mise à jour du chapitre <i>Comprendre les informations de la Marketplace</i>.</li></ul>
2.3	Lyra Collect	27/05/2021	<ul style="list-style-type: none"><li>Ajout du chapitre <i>Paiement par vouchers</i> et ses sous-chapitres</li><li>Suppression du chapitre <i>Paiement par Titres-Restaurant Dématérialisés</i></li><li>Ajout du chapitre <i>Paiement à l'aide d'un lien persistant</i> et ses sous-chapitres</li><li>Mise à jour du chapitre <i>Objet Order</i> dans le dictionnaire de données</li><li>Ajout du chapitre <i>Objet Order voucher</i> dans le dictionnaire de données</li></ul>
2.2	Lyra Collect	13/04/2021	Ajout de précisions dans le chapitre <i>Méthodes d'envoi</i> .
2.1	Lyra Collect	28/01/2021	<ul style="list-style-type: none"><li>Mise à jour de la description de l'objet <i>Order</i>.</li><li>Mise à jour du chapitre <i>Traiter le retour vers la Marketplace</i>.</li><li>Ajout du support du SDK mobile dans le chapitre <i>Les Ressources du Webservice</i>.</li></ul>
2.0	Lyra Collect	21/12/2020	<ul style="list-style-type: none"><li>Ajout du chapitre <i>Comprendre les informations de la Marketplace</i>.</li><li>Ajout du chapitre <i>Comprendre les cashouts</i>.</li><li>Ajout du chapitre <i>Cinématique du cashout</i>.</li><li>Ajout du chapitre <i>Identifier le cashout et les commandes associées</i>.</li><li>Mise à jour du chapitre <i>Liste des cashouts</i> : précision dans "capture_label".</li></ul>
1.9	Lyra Collect	23/11/2020	<ul style="list-style-type: none"><li>Mise à jour du chapitre <i>Analyser le résultat du paiement</i>.</li></ul>
1.8	Lyra Collect	17/11/2020	<ul style="list-style-type: none"><li>Ajout du chapitre <i>Générer un client par OpenAPI</i>.</li><li>Ajout des valeurs possibles pour l'attribut <code>auto_code</code> de l'objet <b>Transaction</b>.</li><li>Mise à jour du chapitre <i>Paiement en validation manuelle</i>.</li></ul>
1.7	Lyra Collect	11/08/2020	<ul style="list-style-type: none"><li>Ajout de la description de l'objet <i>Transaction</i>.</li><li>Correction de la formule de la commission minimale.</li><li>Correction de l'URL d'appel dans l'exemple de remboursement.</li></ul>
1.6	Lyra Collect	11/06/2020	Ajout du chapitre <i>Paiement par Titres-Restaurant Dématérialisés</i> .
1.5	Lyra Collect	28/05/2020	<ul style="list-style-type: none"><li>Ajout des statuts de commande autorisant l'annulation.</li></ul>

Version	Auteur	Date	Commentaire
			<ul style="list-style-type: none"> <li>Mise à jour des statuts de commande autorisant le remboursement.</li> <li>Ajout du calcul de la commission minimale.</li> <li>Ajout du chapitre <i>Paiement à l'initiative du marchand</i>.</li> </ul>
1.4	Lyra Collect	06/05/2020	<ul style="list-style-type: none"> <li>Ajout de la création des paiements en N fois.</li> <li>Ajout de la modification d'une commande réglée en N fois.</li> <li>Mise à jour du principe de commission.</li> <li>Mise à jour de la ressource items.</li> <li>Mise à jour du chapitre <i>Réaliser un paiement</i>.</li> <li>Correction de l'URL d'appel POST ORDER dans les exemples.</li> </ul>
1.3	Lyra Collect	16/07/2019	<p>Ajout de la ressource <b>Cashouts</b>.  Mise à jour des ressources <b>Items</b>, <b>Refunds</b> et <b>Order</b>.  Objet <b>Order</b>: ajout de l'attribut <b>form_token</b>.  Ajout des actions:</p> <ul style="list-style-type: none"> <li>Lister les cashouts d'une marketplace,</li> <li>Obtenir le détail d'un cashout,</li> <li>Modifier une demande de remboursement,</li> <li>Annuler une demande de remboursement,</li> <li>Débloquer le paiement d'un item préalablement bloqué.</li> </ul> <p>Mise à jour des chapitres:</p> <ul style="list-style-type: none"> <li>Suivre l'évolution de la demande de remboursement,</li> <li>Modifier une commande,</li> <li>Présentation du Webservice,</li> <li>Les grandes étapes du paiement,</li> <li>Comprendre le déroulement d'un paiement marketplace,</li> <li>Réaliser un paiement.</li> </ul> <p>Ajout des chapitres:</p> <ul style="list-style-type: none"> <li>Déclenchement manuel du paiement des items,</li> <li>Consulter les cashouts de la marketplace,</li> <li>Modifier une demande de remboursement,</li> <li>Annuler une demande de remboursement,</li> <li>Paiement en préautorisation (validation manuelle).</li> </ul>
1.2	Lyra Collect	22/01/2019	<p>Ajout de la définition du terme "<b>order</b>".  Ajout de la description du <b>Principe de commission</b>.  Ajout des ressources <b>Tokens</b> et <b>Refunds</b>.  Ajout du cycle de vie des ressources <b>Tokens</b> et <b>Refunds</b>.  Ajout de l'action <b>Annuler une commande</b>.  Modification du chapitre <b>Réaliser un paiement</b>:</p> <ul style="list-style-type: none"> <li>exemples de code mis à jour,</li> <li>ajout de l'utilisation de la ressource Tokens.</li> </ul> <p>Ajout des chapitres:</p> <ul style="list-style-type: none"> <li>Analyser le résultat d'une demande de token.</li> <li>Récupérer le détail d'un token</li> <li>Modifier un order</li> <li>Paiement avec délai de capture</li> <li>Objet Token</li> </ul>

Version	Auteur	Date	Commentaire
			<ul style="list-style-type: none"> <li>• Objet Alias</li> <li>• Objet Refund</li> </ul>
1.1	Lyra Collect	12/07/2018	Précisions sur les chapitres : ajout de noms dans les objets Suppression de la ressource transfers Ajout de chapitres : <ul style="list-style-type: none"> <li>• Consulter les sous-marchands enregistrés dans la marketplace</li> <li>• Rembourser un paiement</li> </ul>
1.0	Lyra Collect	12/03/2015	Version initiale

Ce document et son contenu sont strictement confidentiels. Il n'est pas contractuel. Toute reproduction et/ou distribution de tout ou partie de ce document ou de son contenu à une entité tierce sont strictement interdites ou sujettes à une autorisation écrite préalable de Lyra Collect. Tous droits réservés.

## 2. GLOSSAIRE DE LA MARKETPLACE

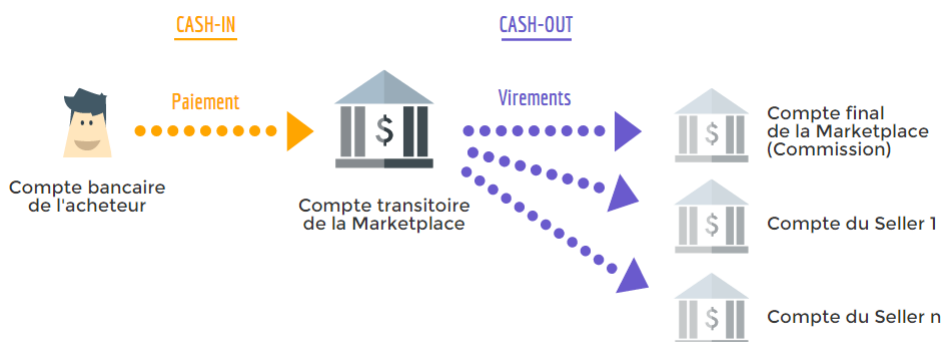
---

Vocabulaire	Description
Cash-in	Paiement de l'acheteur à la Marketplace (CB, VISA...).
Cash-out	Virement de la Marketplace vers les Sellers.
Marketplace	Place de marché (site Internet proposant des articles de divers vendeurs/prestataires)
Order	Un order est composé d'items, il correspond à une commande de la Marketplace. Chaque item est rattaché à un seller.
Seller	Vendeur sur la Marketplace, aussi appelé prestataire. Peut être une société (titulaire d'un SIRET) ou un particulier.

### 3. PRÉSENTATION DE LA MARKETPLACE

La plateforme de paiement propose des services de paiement pour des Marketplaces (places de marché), qui vendent des produits ou des services sur Internet pour le compte de vendeurs avec lesquels elles ont signé un accord commercial en ce sens.

Principe des flux bancaires :



Vocabulaire	Description
Cash-in	Paieement de l'acheteur à la Marketplace (CB, VISA...).
Cash-out	Virement de la Marketplace vers les Sellers.

Les paieements des acheteurs sont ventilés vers les comptes de paieements des sellers. La plateforme de paieement déclenche ensuite des virements vers les comptes bancaires des Sellers de la Marketplace.

### Principe de commission

La commission représente la part du montant de la commande qui revient à la Marketplace.

**Remarque :**

C'est sur cette commission qu'est déduite la part affectée à Lyra Collect. Par conséquent, la commission de la Marketplace doit permettre de couvrir les frais de Lyra Collect. Pour cette raison, le montant global des commissions comprises dans la commande est contrôlé avant le paieement de la commande.

### Calcul de la commission minimale

Pour éviter de soulever une erreur lors contrôle de la commission, vous pouvez préalablement vérifier que le montant de la commission envoyé à l'API marketplace est supérieur ou égal à la commission minimale, dont la formule est donnée ci-dessous :

Symbole	Définition	Exemple 1	Exemple 2
$Com_{min}$	Commission minimale		
$a$	<b>Coefficient</b> de la commission minimale au prorata du montant total de la commande (commission incluse)	2%	1%
$b$	Commission fixe <b>par transaction</b>	0,50€	0,20€



Symbole	Définition	Exemple 1	Exemple 2
$M$	Montant <i>hors commission</i> de la commande	100,00€	100,00€
$n$	Nombre de transactions de la commande	1	3
$tva$	Taux de TVA	20%	20%

Remarque :

- $a$ ,  $b$  et  $tva$  sont paramétrés au niveau de l'API marketplace. Si vous ne les connaissez pas, renseignez-vous auprès de votre contact commercial.
- De son côté, la marketplace qui transmet l'order à l'API marketplace maîtrise  $M$  et  $n$ .

La commission minimale s'obtient par la formule suivante en fonction du montant de la commande hors commission ( $M$ ) :

$$Com_{min}(M) = \frac{aM + bn}{\frac{1}{1 + tva} - a}$$

Avec les montants donnés en exemple, exprimés en centimes, on a donc une commission minimale de :

Exemple 1 :

$$Com_{min}(M) = \frac{10000 \times 0.02 + 50}{\frac{1}{1 + 0.2} - 0.02} = 307.37$$

Exemple 2 :

$$Com_{min}(M) = \frac{10000 \times 0.01 + 3 \times 20}{\frac{1}{1 + 0.2} - 0.01} = 194.33$$

Le montant de la commission transmis devra donc être au moins égal à 3,08€ dans l'exemple 1, et de 1,95€ dans l'exemple 2.

## Méthodes d'envoi

Deux méthodes permettent de renseigner le montant de la commission :

1. **la commission est définie au niveau de la commande** : le montant de la commission est indiqué à travers un item de type commission, marqué par l'attribut **is\_commission=true**. Le seller de cet item doit être le seller Marketplace, c'est-à-dire celui dont l'attribut **is\_marketplace\_seller** vaut **"true"**. Dans ce cas, le montant de la commission s'ajoute au montant des autres items.

Exemple :

```
"items": [
  {
    "seller": "4d20a9d4-0526-4474-b452-e936dc25418d",
    "reference": "produit_sous_marchand",
    "description": "Produit",
    "amount": 10000,
    "quantity": 1
  },
  {
    "seller": "72ccc2ff-b455-4653-847e-deb6fee99f8d",
    "reference": "marketplace_commission",
    "description": "Commission",
    "amount": 1000,

```

```
"quantity": 1,  
  "is_commission": true  
}]
```

Dans cet exemple, si la devise de la commande est EUR et qu'elle contient 2 items, dont 1 item de 100€, plus 1 item de 10€ de commission, alors le montant total de la commande est de  $100 + 10 = 110$ €.

Le seller recevra 100€ et la marketplace recevra 10€ (moins la part affectée à Lyra Collect).

- 2. La commission est définie au niveau d'un item :** le montant de la commission est indiqué en valorisant l'attribut **commission\_amount** de l'item. Dans ce cas, le montant de la commission vient en déduction du montant associé à l'item et le **commission\_amount** peut ainsi être défini sur les sous-marchands (et non sur l'opérateur marketplace).

Exemple :

```
{  
  "seller": "4d20a9d4-0526-4474-b452-e936dc25418d",  
  "reference": "abcdef",  
  "description": "Restaurant",  
  "amount": 10000,  
  "quantity": 1,  
  "commission_amount": 1000,  
  "is_commission": false  
}
```

Dans cet exemple, si la devise de la commande est EUR, l'item vaut 100€ et l'attribut **commission\_amount** est renseigné à 10€, la Marketplace percevra 10€, et le marchand de l'item recevra  $100 - 10 = 90$ €.

Cette modalité est utile pour maîtriser la répartition des commissions entre les différents items (et donc entre les différents sellers).

**Notes :**

- Il est possible de combiner les deux modes, c'est-à-dire de définir un **commission\_amount** dans un ou plusieurs items, et d'ajouter un item de type commission. Dans ce cas, les commissions définies au niveau de la commande s'ajoutent aux commissions définies au niveau des items.
- En revanche, il ne fait pas sens – et il n'est pas possible – de définir un **commission\_amount** sur un item de type commission.
- La valeur par défaut de l'attribut **is\_commission** est "**false**". Il n'est donc pas nécessaire de l'indiquer pour les items non-commission.

Un exemple de création d'order est donné au chapitre [Réaliser un paiement](#) à la page 25.

## 4. 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.

### 4.1. 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 réussit, 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**.

### 4.2. Les Ressources du Webservice

Le Webservice Marketplace est disponible à l'adresse :

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

Le Webservice propose des ressources accessibles par des méthodes HTTP différentes:

Ressources	Action	Méthode HTTP	URI
<b>Marketplaces</b>	Récupérer la liste des Marketplaces	GET	/marketplaces/
	Récupérer les informations de la Marketplace	GET	/marketplaces/{id_marketplace}
<b>Sellers</b>	Récupérer un prestataire	GET	/sellers/{id_seller}
	Récupérer la liste des prestataires	GET	/marketplaces/{id_marketplace}/sellers
<b>Orders</b>	Créer une commande	POST	/orders?expand=items
	Modifier une commande	PUT	/orders/{id_order}
	Récupérer une commande	GET	/orders/{id_order}
	Récupérer la liste des commandes de la Marketplace	GET	/orders
	Finaliser une commande et préparer la page de paiement	GET	/orders/{id_order}/execute
	Finaliser une commande et récupérer le token du formulaire embarqué	GET	/orders/{id_order}/embedded-execute
	Finaliser une commande et récupérer le token du formulaire initialiser le paiement depuis le SDK mobile	GET	/orders/{id_order}/embedded-execute

Ressources	Action	Méthode HTTP	URI
	Annuler une commande	DELETE	/orders/{id_order}
<b>Items</b>	Récupérer un élément d'une commande	GET	/items/{id_item}
	Récupérer les éléments d'une commande	GET	/orders/{id_order}/items
	Récupérer les éléments d'un prestataire	GET	/sellers/{id_seller}/items
	Créer un élément (article) dans une commande	POST	/orders/{id_order}/items
	Débloquer le paiement d'un item préalablement bloqué	POST	/items/{id_item}/activate
<b>Refunds</b>	Créer un remboursement	POST	/refunds
	Récupérer les éléments d'un remboursement	GET	/refunds/{id_refund}
	Récupérer la liste des remboursements d'une commande	GET	/{id_order}/refunds
	Modifier un remboursement	PUT	/refunds/{id_refund}/
	Supprimer un remboursement	DELETE	/refunds/{id_refund}/
<b>Tokens</b>	Demande de création / modification de token	POST	/tokens
	Récupérer une demande de token simple	GET	/tokens/{id_token}
	Récupérer une demande de token liée à un order	GET	/tokens/{id_order}
<b>Alias</b>	Récupérer les éléments d'un alias	GET	/marketplaces/ {id_marketplace}/alias/ {id_alias}
<b>Cashouts</b>	Lister les cashouts vers les sellers d'une marketplace	GET	/cashouts
	Obtenir le détail d'un cashout	GET	/cashouts/{id_cashout}

### 4.3. Prérequis

---

Contactez Lyra Collect pour activer l'accès à la Marketplace et obtenir l'environnement.

#### Pré requis pour la Marketplace

- Souscrire à l'offre Marketplace.

En souscrivant à cette offre, la plateforme de paiement vous envoie en retour des données indispensables pour accéder à la Marketplace :

- l'**ID** de votre Marketplace
- un **login** et un **mot de passe** indispensables pour vous authentifier.

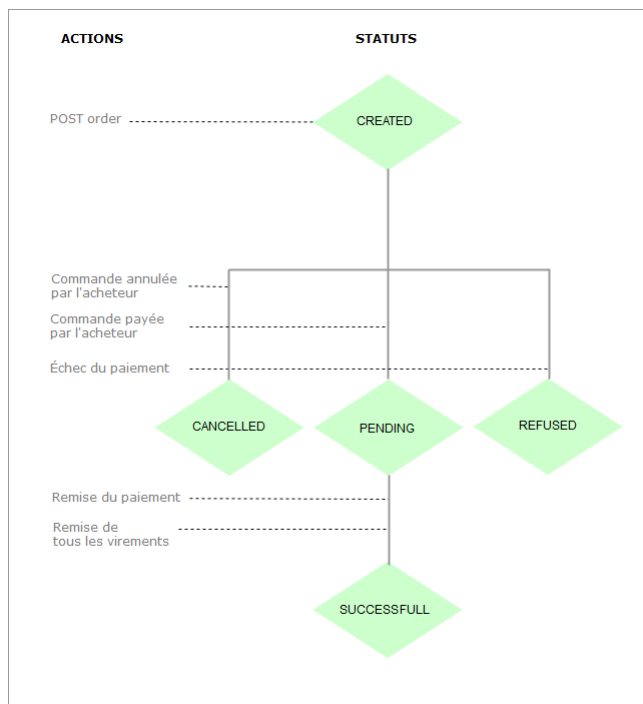
## 5. COMPRENDRE LE DÉROULEMENT D'UN PAIEMENT MARKETPLACE

Ci-dessous la cinématique du déroulement d'un paiement sur la Marketplace :

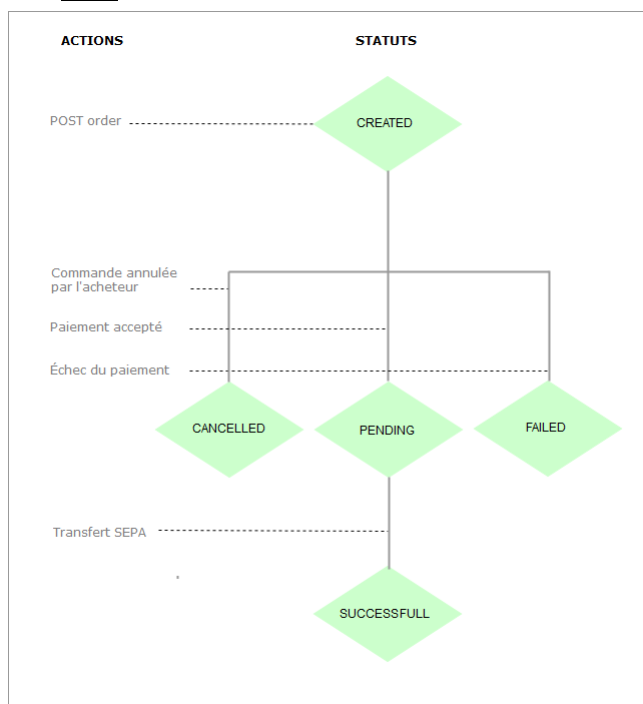
Étapes	Acteurs	Actions
1	Acheteur	Valide son panier sur le site de la Marketplace.
2	Marketplace	Créé un Order via l'API REST (POST ORDER) contenant des items.
3	Marketplace	Figé l'Order via l'API REST (GET ORDER). Finalise l'Order et prépare la page de paiement ou le token du formulaire embarqué.
4	Plateforme de paiement	Créé le contexte de paiement de l'Order. Si le formulaire embarqué est activé, retourne à la marketplace un token (form-token) à insérer dans le formulaire embarqué. Sinon, retourne à la Marketplace une URL pour rediriger l'acheteur vers la page de paiement.
5	Marketplace	Si le formulaire embarqué est activé, construit le formulaire sur la page du site marchand. Sinon, redirige l'acheteur vers l'URL donnée par la plateforme de paiement récupérée par le GET ORDER.
6	Acheteur	Saisit ses informations bancaires (+ authentification 3D Secure sur les pages de paiement).
7	Plateforme de paiement	Valide et enregistre la transaction.
8	Plateforme de paiement	Met à jour le statut des Items et de l'Order.
9	Plateforme de paiement	Envoie une notification à la Marketplace (et éventuellement au formulaire embarqué) indiquant que le statut de l'Order a changé.
10	Marketplace	Appelle la plateforme de paiement pour connaître le statut de l'Order et met à jour son système (GET ORDER).
11	Plateforme de paiement	Redirige l'acheteur vers la Marketplace (URL de retour donnée par la Marketplace = site marchand).
12	Plateforme de paiement	Effectue la remise en banque de la transaction par carte bancaire (ou autre). Créé les virements sur la boutique du Seller. Envoie le fichier d'ordres de virement à la banque.
13	Plateforme de paiement	Notifie la Marketplace du changement de statut de l'Order et des Items.
14	Marketplace	Appelle la plateforme de paiement pour connaître le statut de l'Order et met à jour son système (GET ORDER).
15	Marketplace	Valide manuellement le fichier d'ordres de virement sur l'interface de sa banque.
16	Banque	Exécute les virements demandés.

Le statut des ressources varie tout au long du déroulement d'un paiement.

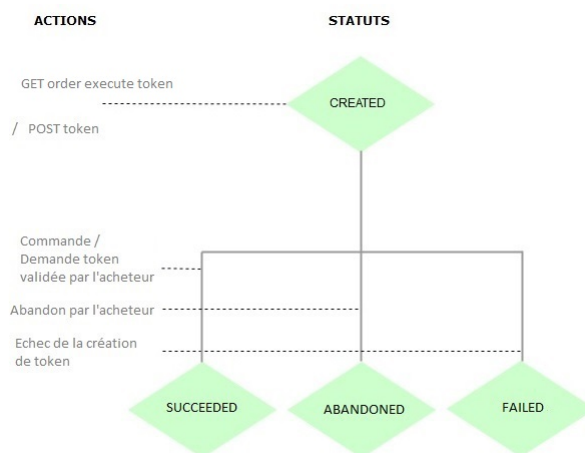
- Cycle de vie de la ressource **Order**



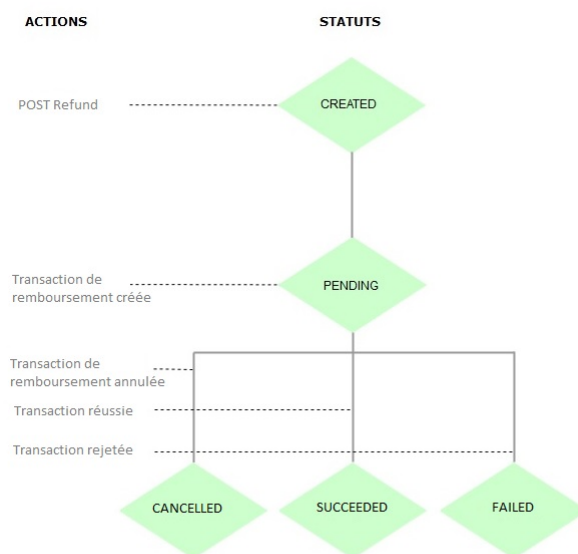
- Cycle de vie de la ressource **Item**



- Cycle de vie de la ressource Tokens



- Cycle de vie de la ressource Refunds



## 6. S'IDENTIFIER LORS DES ÉCHANGES

L'identification s'effectue au moyen d'un en-tête HTTP.

La méthode utilisée est **HTTP Basic Authentication**.

Dans chaque requête HTTP, l'en-tête doit contenir les informations permettant à la Marketplace de s'authentifier auprès du Webservice Marketplace.

Description des en-têtes HTTP :

En-têtes	Description	Valeurs à utiliser
<b>Accept</b>	Détermine le format du contenu qui sera retourné par le serveur. Architecture REST qui permet de faire des échanges au format json	'Accept: application/json'
<b>Authorization</b> <b>Description</b>	: Jeton d'identification selon le principe de "Basic Authentication over HTTPS". Se compose de l'identifiant et du mot de passe de l'utilisateur de l'API séparés par le caractère ":", le tout encodé en base 64.	'Authorization: Basic YWRtaW46YWRtaW4='
<b>Content-type</b>	Détermine le format du contenu qui est envoyé au serveur.	'content-type: application/json'
<b>Method</b>	Voir tableau au chapitre <b>Les Ressources du Webservice</b> pour connaître la méthode à utiliser en fonction de la ressource.	<b>GET   POST   PUT   DELETE</b>

Les étapes pour construire ces en-têtes sont les suivantes :

1. Utilisez la méthode **Basic Authentication**.
2. Spécifiez dans l'en-tête **Authorization** la méthode utilisée : **Basic** suivie de la représentation en Base 64 du login et du mot de passe séparés par le caractère ":".
3. Encodez le résultat obtenu en base 64.
4. Ajoutez la chaîne en "Basic".

**Remarque :**

*Ne pas oublier de mettre un espace après **Basic**.*

- Exemple **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
```

- Exemple de requête complète en **Python** :

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

- Exemple de requête en **.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;
```



## 7. COMPRENDRE LES INFORMATIONS DE LA MARKETPLACE

A l'aide des identifiants de connexion et de l'identifiant unique de votre Marketplace (`uuid`) communiqués par nos services, vous accédez aux informations de votre Marketplace par un appel à la ressource :

```
GET /marketplaces/{uuid}
```

Ci-dessous, un exemple de réponse, pour une Marketplace dont l'`uuid` est `2434c0a2-9d46-4e96-9553-1536c898625b` :

### Requête

```
GET https://secure.lyra.com/marketplace/marketplaces/2434c0a2-9d46-4e96-9553-1536c898625b
```

### Réponse

```
{
  "href": "https://secure.lyra.com/marketplace/marketplaces/2434c0a2-9d46-4e96-9553-1536c898625",
  "uuid": "2434c0a2-9d46-4e96-9553-1536c898625b",
  "created_at": "2017-03-13T14:58:40.801000Z",
  "updated_at": "2020-12-07T10:41:12.184969Z",
  "reference": "MKP000001",
  "description": "La maison du cheesecake",
  "billing_method": "CASHOUT",
  "bic": "",
  "iban": "",
  "vads_key": "12345678",
  "vads_cert": "123456789012345",
  "status": "ACTIVE",
  "links": {
    "sellers": {
      "href": "https://secure.lyra.com/marketplace/marketplaces/2434c0a2-9d46-4e96-9553-1536c89"
    },
    "orders": {
      "href": "https://secure.lyra.com/marketplace/marketplaces/2434c0a2-9d46-4e96-9553-1536c898"
    },
    "registrations": {
      "href": "http://secure.lyra.com/marketplace/marketplaces/2434c0a2-9d46-4e96-9553-1536c8986"
    },
    "webhooks": {
      "href": "http://secure.lyra.com/marketplace/marketplaces/2434c0a2-9d46-4e96-9553-1536c8986"
    }
  },
  "max_capture_delay": 6,
  "tva_rate": "20.00",
  "currencies": [
    {
      "currency": "EUR",
      "commission_prorata": 2.0,
      "commission_fix": 20,
      "is_active": true
    },
    {
      "currency": "GBP",
      "commission_prorata": 1.0,
      "commission_fix": 60,
      "is_active": true
    }
  ],
  "vouchers": [
    {
      "contract_type": "CONECS"
    }
  ]
}
```

Que peut-on comprendre de cet exemple?

1. Tout d'abord, que la Marketplace est belle et bien active. Elle correspond par ailleurs à l'identifiant boutique "12345678" disposant de la clé "123456789012345".

Ces deux informations sont accessibles depuis le Back Office Expert et sont renseignées directement par les services de Lyra Collect lors de l'enregistrement de la Marketplace.

2. Son `billing_method` indique qu'elle est paramétrée pour un prélèvement dit "cashout". Par opposition avec un prélèvement "mensuel", la marketplace doit donc respecter un montant minimal de commission à chaque commande.
3. La listes des webhooks accessibles et utilisés par la marketplace se trouvent à l'adresse suivante: `""https://secure.lyra.com/marketplace/marketplaces/2434c0a2-9d46-4e96-9553-1536c898625b/webhooks"`

4. La liste des vendeurs est accessible à l'adresse suivante :

```
"https://secure.lyra.com/marketplace/marketplaces/2434c0a2-9d46-4e96-9553-1536c898625b/sellers"
```

5. La liste des commandes est accessible à l'adresse suivante :

```
"http://secure.lyra.com/marketplace/marketplaces/2434c0a2-9d46-4e96-9553-1536c898625b/orders"
```

6. Deux devises sont activées sur la Marketplace, avec des paramètres de commissions minimales distincts :
  - L'**Euro**, avec 2% du prorata du montant total de la commande et 10 centimes par transaction;
  - La **Livre sterling**, avec 1% du montant total de la commande et 60 pence.
7. Et pour finir, la marketplace peut admettre l'enregistrement de contrats acquéreurs 'CONECS' sur ses sous-marchands, lesquels peuvent alors proposer le *[paiement par ticket restaurant](#)*.

## 8. COMPRENDRE LES CODES RETOUR DE STATUT HTTP ENVOYÉS PAR LE WEBSERVICE

---

Code erreur	Description
200 - 20X	Requête traitée avec succès
401	Accès non autorisé (problème dans le header Authorization)
403	Accès interdit (L'utilisateur de l'API n'a pas la permission d'effectuer cette requête)
400	Mauvaises données d'entrée (exemple : le format de certains paramètres n'est pas respecté)
404	L'objet demandé est introuvable (On a demandé un uuid qui n'existe pas)
500	Erreur interne du serveur

Ci-dessous quelques pistes d'analyse pour vous aider à résoudre rapidement les problèmes survenus :

### Erreur 400 :

- **"?expand=items"** est absent à la fin de la requête POST order
- les accents ne sont pas encodés en ANSI
- la référence d'un item comporte des espaces.

## 9. CONSULTER LES SOUS-MARCHANDS ENREGISTRÉS DANS LA MARKETPLACE

Pour consulter les sous-marchands enregistrés dans la marketplace, vous devez lancer l'appel en mode GET.

```
GET /marketplaces/{marketplace}/sellers
```

Remplacez {marketplace} par l'uuid fourni par Lyra Collect.

Par exemple :

Requête

```
GET https://secure.lyra.com/marketplace/marketplaces/30805a03-11ec-4447-93a5-243f39c89009/sellers
```

Réponse

```
{
  "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. UTILISER DES WEBHOOKS

Les webhooks vous permettent d'être automatiquement notifié lors du changement de statut d'un objet (commande, enregistrement, remboursement, etc.). En suivant son cycle de vie, vous êtes ainsi à même de réagir en déclenchant de nouveaux événements, de nouveaux appels, etc.

**Typiquement: un webhook vous parvient lorsque les fonds d'une commande sont remisés sur le compte de règlement, à partir de quoi vous pouvez, si cela est convenu ainsi dans votre processus d'affaires, envoyer une information à votre sous-marchand pour valider la livraison.**

Pour des raisons de sécurité, le webhook est réduit à sa plus simple expression, inexploitable en l'absence d'un accès authentifié à l'API. Il s'agit d'une requête POST dont le corps s'exprime de la manière suivante :

```
{"order": "dd5e4c4c-2c07-4af8-ae30-15f4c6d5b5e5" }
```

Il vous appartient, à la réception de ce webhook, d'interroger l'objet en question depuis la ressource correspondante mise à votre disposition, et de prendre ainsi connaissance du nouveau statut.

Exemple :

```
GET https://secure.lyra.com/orders/dd5e4c4c-2c07-4af8-ae30-15f4c6d5b5e5)
```

En temps normal, le webhook parvient au serveur de marketplace quelques secondes après la mutation de l'état. En cas d'erreur à l'envoi (réseau indisponible, par exemple), il peut être réexpédié jusqu'à deux fois supplémentaires, soit, trois fois en tout, à 15 minutes d'intervalle.

Le webhook n'est pas répété en cas d'erreur à la réception (par exemple: si votre serveur renvoie un statut 400, 404, 500, etc.). Pour éviter les erreurs les plus courantes, un contrôle d'envoi et de réception est effectué au moment de l'enregistrement du webhook.

Cette section vous présente donc comment :

1. définir un (ou plusieurs) point d'accès;
2. accéder aux webhooks accessibles, et ceux actuellement définis sur votre marketplace;
3. enregistrer, modifier et supprimer vos webhooks.

### 10.1. Définir l'adresse du webhook

Un webhook suppose que vous mettiez à disposition de l'API une adresse sur laquelle recevoir une information.

Comme nous l'avons vu, les webhooks sont définis par type d'objet. Vous pouvez donc définir autant de points d'accès que d'objets à suivre.

Ces points d'accès doivent être capables de recevoir et de traiter des appels en POST avec l'en-tête `Content-Type: application/json`, et le contenu suivant :

```
{"<event_type>": "<uuid>"}
```

Par exemple :

```
{"order": "dd5e4c4c-2c07-4af8-ae30-15f4c6d5b5e5" }
```

Actuellement, les objets concernés sont les suivants :

- **order**
- **token**
- **registration**
- **refund**

A chaque création ou modification d'un webhook, un test de notification est réalisé avec un corps de requête vide auquel votre serveur doit répondre avec un statut 200 (voir chapitre [Enregistrer, modifier ou supprimer un webhook](#) à la page 23).

Et pour accroître la sécurité de votre processus, vous pouvez contrôler que le webhook émane bien de la plage d'IP **194.50.38.0/24**.

## 10.2. Accéder aux webhooks définis et accessibles sur votre marketplace

Tous vos webhooks sont accessibles depuis la ressource :

```
GET /marketplaces/{marketplace}/webhooks
```

La ressource renvoie :

1. la liste des webhooks actuellement définis sur la marketplace,
2. et la liste des webhooks accessibles, mais non utilisés.

**Note** : cette liste est susceptible de s'élargir en fonction des développements à venir de l'API Marketplace

Exemple :

Requête

```
GET https://secure.lyra.com/marketplace/marketplaces/6f6b04c2-0e99-4f8d-b710-8856f5654bb8/webhooks
```

Réponse

```
[
  {
    "event_type": "order",
    "target": "https://mymarketplace.com/mkp/webhooks/order.php"
  },
  {
    "event_type": "token",
    "target": "https://mymarketplace.com/mkp/webhooks/order.php"
  },
  {
    "event_type": "refund",
    "target": null
  },
  {
    "event_type": "registration",
    "target": null
  }
]
```

Cette réponse indique qu'une seule adresse de webhook est utilisée sur deux objets, `order` et `token`, mais que les webhooks pour les objets `refund` et `registration` ne sont pas renseignés, et sont donc inactifs.

## 10.3. Enregistrer, modifier ou supprimer un webhook

---

### 10.3.1. Ajouter un webhook

Pour ajouter un webhook, vérifiez d'abord que l'adresse du webhook est bien publiquement exposée et qu'elle renvoie un statut 200 à la réception d'une requête vide, puis exécutez une requête

```
POST /marketplaces/{marketplace}/webhooks
```

avec le corps suivant :

```
{
  "event_type": "<event_type>",
  "target": "<webhook_url>"
}
```

Par exemple :

Requête

```
POST https://secure.lyra.com/marketplaces/6f6b04c2-0e99-4f8d-b710-8856f5654bb8/webhooks
```

Corps

```
{
  "event_type": "registration",
  "target": "https://mymarketplace.com/mkp/webhooks/sellers.php"
}
```

En cas d'erreur lors du test, la requête de création (ou de modification) renverra une erreur 400.

Exemple si vous avez déclaré une page inexistante comme webhook :

```
{"error": "Url https://mymarketplace.com/mkp/webhooks/sellers.php returned a status 404 instead of 200"}
```

En cas de succès, le serveur renvoie un code 200 et l'objet à l'identique.

### 10.3.2. Modifier un webhook

Les mêmes principes s'appliquent à la modification du webhook, si ce n'est qu'il faut effectuer une requête `PUT` au lieu de `POST`.

```
PUT /marketplaces/{marketplace}/webhooks
```

Exemple pour attribuer une nouvelle adresse au webhook existant sur l'objet `token` :

Requête

```
PUT https://secure.lyra.com/marketplaces/6f6b04c2-0e99-4f8d-b710-8856f5654bb8/webhooks
```

Corps

```
{
  "event_type": "token",
  "target": "https://mymarketplace.com/mkp/webhooks/token.php"
}
```

### 10.3.3. Supprimer un webhook

Pour supprimer un webhook, il suffit d'envoyer une requête

```
DELETE /marketplaces/{marketplace}/webhooks/{event_type}
```

Par exemple pour supprimer le webhook de l'objet `registration` :

```
DELETE https://secure.lyra.com/marketplaces/6f6b04c2-0e99-4f8d-b710-8856f5654bb8/webhooks/
registration
```

En cas de succès, le serveur renvoie un statut 204.



## 11. RÉALISER UN PAIEMENT

Pour réaliser un paiement, vous devez suivre les étapes suivantes :

1. Créer une commande en faisant un POST sur la ressource ORDER.
2. Modifier éventuellement la commande pour mettre à jour le panier ou les montants des commissions par exemple.
3. Figurer la commande pour effectuer le paiement en faisant un GET :
  - soit sur l'url de l'attribut **execute** fourni dans les links de l'ORDER, dans ce cas la réponse contient l'url de redirection (attribut **payment\_url**),
  - soit sur l'url de l'attribut **execute-embedded** pour initialiser le paiement par formulaire embarqué, et dans ce cas la réponse contient un formToken (attribut **form\_token**).
4. Rediriger l'acheteur vers l'URL de la page de paiement ou afficher le formulaire embarqué en utilisant le form\_token.

### Étape 1

Pour déclarer des items dans votre commande, vous **devez** mentionner l'argument **?expand=items** dans l'url (/marketplace/orders?expand=items).

Dans le cas contraire, vos items ne seront pas pris en compte (la liste d'items retournée par le serveur est vide) et la commande sera créée sans items. Lors de l'exécution de la commande (étape 3) vous rencontrerez une erreur 400 {"amount":"The sum of item amounts cannot be zero or negative."}.

### Exemple d'appel POST ORDER :

```
POST https://secure.lyra.com/marketplace/orders?expand=items
```

```
{
  "marketplace": "ebfb36ab-2d30-4326-adb9-e16b0c9a89f3",
  "reference": "Exemple MKP BURO",
  "description": "Commande fournitures",
  "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": "Stylos bleus",
      "amount": 1000,
      "is_commission": false
    }
  ],
  "buyer": {
    "type": "PRIVATE",
    "email": "jean.dupont@acheteur.fr",
    "phone_number": "33 (0)1 11 22 33 44",
    "reference": "ZhIsYM"
  },
  "shipping": {
    "shipping_method": "RELAY_POINT"
  }
}
```

## Exemple de retour du POST ORDER :

```
{
  "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": "Exemple MKP BURO",
  "description": "Commande fournitures",
  "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": "Stylos bleus",
      "type": "ENTERTAINMENT",
      "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": "ENTERTAINMENT",
      "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
}

```

## Étape 2

Une commande peut être modifiée par une requête **PUT ORDER** (/marketplace/orders/<order\_uuid>?expand=items) à l'aide de l'**order\_uuid** retourné à l'étape 1, et ce, plusieurs fois.

Les attributs **buyer** et **shipping** doivent obligatoirement être renseignés.

La liste des items est renouvelée à chaque PUT. Si vous souhaitez ajouter un item supplémentaire, vous devez le transmettre avec l'ensemble des items déjà créés. Sinon la liste ne contiendra que cet item.

A contrario pour supprimer un item il suffit de renvoyer la liste complète des items sans celui que vous souhaitez supprimer.

Exemple d'appel PUT ORDER pour ajouter un item dans le panier:

```

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

```

```

{
  "marketplace": "ebfb36ab-2d30-4326-adb9-e16b0c9a89f3",
  "reference": "Exemple MKP BURO",
  "description": "Commande fournitures",
  "currency": "EUR",
  "url_return": "https://URLreturn.com",
  "items": [
    {
      "seller": "d0f80202-0676-4d8d-9247-f455f30aec1b",
      "reference": "commburo",
      "description": "commission BURO",
      "amount": 270,
      "is_commission": true
    },
    {
      "seller": "5d0ef88c-3345-4b33-948e-80e23d553b73",
      "reference": "STB1",
      "description": "Stylos bleus",
      "amount": 1000,
      "is_commission": false
    },
    {
      "seller": "5d0ef88c-3345-4b33-948e-80e23d553b73",
      "reference": "STR3",
      "description": "Stylos rouges",
      "amount": 1000,
      "is_commission": false
    }
  ],
  "buyer": {
    "type": "PRIVATE",
    "email": "jean.dupont@acheteur.fr",
    "phone_number": "33 (0)1 11 22 33 44",
    "reference": "ZhIsYM"
  },
  "shipping": {
    "shipping_method": "RELAY_POINT"
  }
}

```

### Etape 3

A l'exception des paiements en validation manuelle (voir chapitre [Paiement en validation manuelle](#) à la page 36), vous devez vous assurer *avant* d'effectuer la requête d'exécution et de figer la commande que le montant des commissions est supérieur au minimum requis. Si ce n'est pas le cas, la requête d'exécution renverra une erreur de validation 400 avec le message {"commission": "Insufficient amount of commission."}

Vous trouverez des informations sur la commission minimum dans la section Principe de commission du chapitre [Présentation de la Marketplace](#) à la page 8.

#### Exemple d'appel GET via la redirection pour figer la commande :

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

#### Exemple de réponse en GET :

```
{
  'payment_url': 'https://secure.lyra.com/vads-payment/
exec.refresh.a;jsessionid=HgKAcpcvgZQA29qCZDjXw4kS.marketplacevad01?
cacheId=450272311503195000050'
}
```

#### Exemple d'appel GET via la le formulaire embarqué pour figer la commande :

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

#### Exemple de réponse en GET :

```
{"form_token": "fa0yfV2FQuR3aak1SwsgcuZg195eyJhbW91bnQiOjEwOTAwLCJjdXJyZW5jeSI6IkVVUiIsIm
1vZGUiOiJURVNUiwiidmVyc2l2b2I6Mywib3JkZXJJZCI6IjVjNDk1N2E5LWNkOGEtNGJhZC1hMGNiLTgxMT
ZiMWE3ZDdmNiIsInNob3BOYWllIjoIiThlyYSBTTVMiLCJicmFuZFByaW9yaXR5IjpbIkJBTkNPTlRBQ1QiL
CJDQiIsIkUtQ0FSVEVCTEVVRSIsIk1BU1RFUkNBUCkQiLCVGVVVMVCJ9fX0a702"}
```

Note : le form\_token peut peser jusqu'à 8ko

### Etape 4

Dans le cas d'un paiement par redirection, il suffit de rediriger l'acheteur vers le lien fourni. La même adresse peut être utilisée pour un affichage par iframe.

Dans le cas d'un formulaire de paiement embarqué, le **form\_token** doit être inséré dans le formulaire (attribut **kr-form-token**).

Exemple :

```
<body>
<!-- payment form -->
<div class="kr-embedded"
  kr-form-token="fa0yfV2FQuR3aak1SwsgcuZg195eyJhbW91bnQiOjEwOTAwLCJjdXJyZW5jeSI6IkVVUiIsIm
  1vZGUiOiJURVNUiwiidmVyc2l2b2I6Mywib3JkZXJJZCI6IjVjNDk1N2E5LWNkOGEtNGJhZC1hMGNiLTgxMT
  ZiMWE3ZDdmNiIsInNob3BOYWllIjoIiThlyYSBTTVMiLCJicmFuZFByaW9yaXR5IjpbIkJBTkNPTlRBQ1QiL
  CJDQiIsIkUtQ0FSVEVCTEVVRSIsIk1BU1RFUkNBUCkQiLCVGVVVMVCJ9fX0a702">
<!-- payment form fields -->
<div class="kr-pan"></div>
<div class="kr-expiry"></div>
<div class="kr-security-code"></div>

<!-- payment form submit button -->
<button class="kr-payment-button"></button>
<!-- error zone -->
<div class="kr-form-error"></div>
</div>
</body>
```

## 11.1. Paiement en N fois

---

Dans le cas où l'acheteur souhaite étaler ses paiements dans le temps, il peut effectuer un paiement en plusieurs fois.

Lors d'un paiement en N fois, l'ensemble des transactions sont créées le jour du paiement.

Seule la première échéance peut faire l'objet d'une garantie pour le marchand à condition que sa date de présentation demandée soit antérieure à la date de validité de l'autorisation (en fonction du moyen de paiement).

Au moment du paiement, un contrôle est effectué pour vérifier que le moyen de paiement utilisé est valide pendant toute la durée de l'échéancier.

**Ce mode de paiement n'est pas supporté par le formulaire embarqué.**

La liste des transactions est accessible à l'adresse :

```
GET /orders/{order}/transactions
```

Elle apparaît aussi dans la liste des liens ('links') en réponse à un GET ORDER, POST ORDER, ou PUT ORDER.

Notez cependant que les transactions sont générées après le paiement de la commande. Si l'adresse est consultée avant, la liste des transactions est vide.

Pour initialiser un paiement en N fois, il suffit d'ajouter l'attribut **payment\_config** dans la ressource POST ORDER, au moment de la création de la commande.

Exemple d'utilisation de l'attribut **payment\_config** :

```
{
  "marketplace": "9537e049-8862-400a-ae8d-da2ec9ca6051",
  "payment_config": "MULTI:first=30000;count=3;period=30"
  "reference": "cmd00053",
  "description": "Commande",
  "currency": "EUR",
  "url_return": "http://www.my-website.com",
  "language": "fr"
}
```

Notes :

- Par défaut, le paiement est mono-transactionnel. Si l'attribut **payment\_config** n'est pas renseigné au moment de la création de la commande, il prendra la valeur **SINGLE** ;
- Une fois la commande figée, il n'est pas possible de modifier la valeur de **payment\_config**. Toutefois, il reste possible de modifier certains paramètres des transactions en attente (cf. infra) ;
- Le paiement en N fois n'est actuellement pas compatible avec le paiement en cascade (et donc à fortiori pas non plus compatible avec le système Ticket Restaurant Dématérialisé).

Il est possible de définir deux types de paiements échelonnés différents, selon que la marketplace souhaite créer automatiquement un échéancier régulier, ou bien définir les dates et les montants à son gré.

### 11.1.1. Paiements réguliers : MULTI

Dans le cas d'un échéancier régulier, le premier paiement intervient à la date de la création de la commande, et son montant doit être explicitement indiqué. Les montants et les dates des paiements suivants sont calculés en fonction de la différence restant à payer, le nombre et la fréquence des échéances.

**Syntaxe** : la valeur de **payment\_config** doit être préfixée de la chaîne « **MULTI:** » suivie dans l'ordre par les paires clé=valeur suivantes séparées par des « ; » :

- **first=** : montant du premier paiement (exprimé dans la plus petite unité de la devise).
- **count=** : nombre total d'échéances.
- **period=** : intervalle en nombre de jours entre 2 paiements.

Exemple :

Le montant du dernier versement est calculé en fonction du solde restant à payer.

Si un arrondi est nécessaire, il est pris en charge à ce moment-là.

Par exemple, dans le cas d'une commande de 1000 € passée le 15 janvier, le **payment\_config** suivant :

```
{
  ...
  "amount": "100000",
  "payment_config": "MULTI:first=30000;count=4;period=30",
  ...
}
```

correspond à un paiement en 4 fois, comprenant :

- Un premier versement de 300€ le 15 janvier,
- Un second versement de  $(1000 - 300) / 3 = 233,33€$  le 14 février,
- Un troisième versement de 233,33€ le 16 mars,
- Et un dernier versement de 233.34€ le 15 avril.

### 11.1.2. Echancier personnalisé : MULTI\_EXT

Dans le cas d'un échancier personnalisé, tous les montants et les dates de transactions sont spécifiées par la marketplace. Il est ainsi possible de définir des montants variables, et la première transaction peut être ultérieure au jour de la commande.

Il faut cependant respecter certaines contraintes :

- La date de la première échéance ne doit pas être antérieure au jour du paiement.
- Il n'est possible d'enregistrer qu'une transaction par jour.
- Le délai entre la date du paiement et de la dernière transaction ne peut excéder 365 jours.

**Syntaxe** : La valeur de **payment\_config** doit être préfixée de la chaîne « **MULTI\_EXT:** » suivie de la liste des échéances au format date=montant séparées par des « ; ».

Les dates doivent être exprimées au format YYYYMMDD.

Les montants doivent être exprimés dans la plus petite unité de la devise.

Pour reprendre l'exemple précédent mais pour régler cette fois-ci les échéances de la commande au 15 de chaque mois, la valeur de **payment\_config** devra ressembler à :

```
{
  ...
  "amount": "100000",
  "payment_config": "MULTI_EXT:20200115=30000;20200215=23333;20200315=23333;20200415=23334",
  ...
}
```

Note : L'utilisation de la valeur MULTI\_EXT nécessite la souscription de l'option **Paiement en plusieurs fois avancé**.

## 11.2. Paiement avec délai de capture

---

Un délai de capture de la transaction peut être défini lors de la création d'un order. Pour cela, on renseigne l'attribut **capture\_delay** avec le nombre de jours de délai voulu.

Cette valeur doit être comprise entre 0 et 6 :

Exemple :

Requête

```
POST https://secure.lyra.com/marketplace/orders?expand=items
```

Corps

```
{
  "marketplace": "ebfb36ab-2d30-4326-adb9-e16b0c9a89f3",
  "reference": "Exemple MKP BURO",
  "description": "Commande fournitures",
  "currency": "EUR",
  "capture_delay": "3",
  ...
}
```

Pour modifier la date de capture de la transaction associée à un order déjà exécuté ( de statut PENDING), on ne peut pas utiliser l'attribut **capture\_delay**. Il faut passer "**expected\_capture\_date**" avec la date de capture souhaitée, comprise entre aujourd'hui J et J+6. La date est au format "YYYYMMDDHHMMSS".

Exemple :

Requête

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

Corps

```
{
  "marketplace": "ebfb36ab-2d30-4326-adb9-e16b0c9a89f3",
  "reference": "Exemple MKP BURO",
  "description": "Commande fournitures",
  "currency": "EUR",
  "expected_capture_date": "20190425193000",
  ...
}
```



## 11.3. Paiement avec création d'un alias

---

En même temps que l'on crée un order, il est possible de demander l'enregistrement des informations de paiement sous la forme d'un alias qui pourra être réutilisé.

Pour cela, il faut utiliser la commande "**execute/token**" (ou "**execute-embedded/token**") à la place de "execute" lors de la validation de l'order.

Exemple d'appel GET pour figer la commande :

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

Lors de l'exécution de la commande, un objet nommé "token" est créé. **Il a pour uuid celui de l'Order.**

Il est composé de deux éléments:

- un objet "buyer" tel qu'il est défini dans la commande,
- un attribut "alias" dans lequel seront enregistrées les données du moyen de paiement.

*Consultez le chapitre [Gestion des alias](#) à la page 48 pour plus d'informations sur les tokens et les alias.*

Redirigez ensuite l'acheteur vers l'URL transmise dans **payment\_url** pour procéder au paiement avec enregistrement du moyen de paiement.

Après le paiement, un GET Order ne retournera pas l'alias créé.

Pour obtenir l'alias, réalisez un appel GET token avec l'uuid de l'Order.

Consultez le chapitre [Analyser le résultat d'une demande de token](#) à la page 50 pour voir le résultat du Get token.

## 11.4. Paiement par alias

---

Pour utiliser un token dont on a la référence dans un order, on ajoute l'attribut **alias** à la création de l'order.

Exemple :

Requête

```
POST https://secure.lyra.com/marketplace/orders?expand=items
```

Corps

```
{
  "marketplace": "ebfb36ab-2d30-4326-adb9-e16b0c9a89f3",
  "reference": "Exemple MKP BURO",
  "description": "Commande fournitures",
  "currency": "EUR",
  "alias": "1144951ea7ab42989c97159b3dfc0382",
  ...
}
```

En appelant "**execute**" sur cet order, le paiement se fera avec le token qui a pour alias "1144951ea7ab42989c97159b3dfc0382" via la page de paiement.

En appelant "**execute-embedded**" sur cet order, le paiement se fera avec le token qui a pour alias "1144951ea7ab42989c97159b3dfc0382" via le formulaire embarqué.

En appelant "**execute/token**", le token sera également mis à jour avec les informations contenues dans l'order, via la page de paiement.

En appelant "**execute-embedded/token**", le token sera également mis à jour avec les informations contenues dans l'order via le formulaire embarqué.

## 11.5. Paiement à l'initiative du marchand

---

L'utilisation d'un alias lors de la création d'un order permet en outre d'exécuter des paiements à l'initiative du marchand ("Merchant Initiated Transaction") : le paiement s'effectue alors *sans* interaction avec le porteur.

Cette fonctionnalité est notamment utile dans le cas d'abonnements.

Pour ce faire, créez un paiement par alias (voir chapitre [Paiement par alias](#) à la page 34), puis utilisez le lien **execute-mit** retourné par le serveur.

Par exemple :

Requête

```
POST https://secure.lyra.com/marketplace/orders?expand=items
```

Corps

```
{
  "marketplace": "ebfb36ab-2d30-4326-adb9-e16b0c9a89f3",
  "reference": "Exemple MKP BURO",
  "description": "Commande fournitures",
  "currency": "EUR",
  "alias": "1144951ea7ab42989c97159b3dfc0382",
  ...
}
```

Dans la réponse du serveur, l'attribut **links** contient le lien **execute-mit** à appeler.

```
{...
  "links": {
    "execute-mit": {
      "href": "http://mymarketplace.com/marketplace/orders/5a439e70-4ccc-4d86-bf30-223552e2c74f/execute-mit"
    },
    ...
  }
}
```

## 11.6. Paiement en validation manuelle

---

Par défaut, la validation de l'ordre se fait de façon automatique au moment de l'étape 2 du paiement (lors du GET <url>/execute).

Il est également possible de dissocier les étapes d'autorisation de paiement et de remise en banque des transactions.

Cela permet de réaliser des autorisations alors que le dispatching final des items du panier entre montant vendeurs et montant commission marketplace n'est pas encore connu.

### Étape 1


Pour mettre en œuvre cette fonctionnalité, il suffit d'ajouter l'attribut **awaiting\_validation** lors de la création de l'ordre et lui assigner la valeur **true**.

Exemple d'appel POST ORDER de création en validation manuelle :

```
POST https://secure.lyra.com/marketplace/orders?expand=items
```

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

Dans cet exemple, le montant de l'item (montant acheteur) est connu, mais pas sa répartition entre la part sous-marchand et la part de la commission de l'opérateur (la marketplace).

 **Remarque :** Notez toutefois qu'il est possible de préciser dès cette étape les commissions de l'opérateur. Ces dernières pourront, si nécessaire, être mises à jour à l'étape 4.

### Étape 2

L'étape 2 est exécutée de la même manière que pour un paiement standard et l'URL de redirection pour procéder au paiement est retournée.

Exemple d'appel GET pour figer la commande :

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

l'order étant en attente de validation, les contrôles sur le montant minimum de commission ne sont pas exécutés.

### Etape 3

Une fois le paiement effectué avec succès, l'order passe à l'état PENDING et les transactions associées à au statut **TO\_VALIDATE**.

### Etape 4

Cette étape, facultative, permet d'indiquer le montant des commissions de l'opérateur si ces dernières n'ont pas été fournies à l'étape 1.

Il est ainsi possible d'ajouter :

- un item commission
- et/ou des commissions sur les items (cf paragraphe dédié).

Un ou plusieurs items peuvent également être supprimés (fonction standard).

Le montant total de l'order peut être diminué mais il ne peut pas être augmenté.

En revanche, l'attribut **awaiting\_validation** ne peut pas être modifié.

### Etape 5

Le processus se termine par la validation de l'order. C'est lors de cette étape que l'on contrôle que le montant de commission est suffisant.

Pour valider un Order, il faut faire un POST sur l'URL de l'order, attribut **validate**.

Exemple d'appel POST ORDER pour le valider manuellement :

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

## Exemple de retour du POST ORDER valide :

Le détail de l'order est retourné. L'attribut **awaiting\_validation** de l'order prend alors la valeur **false** et les transactions associées passent de l'état **TO\_VALIDATE** à **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": "Exemple MKP BURO 2",
  "description": "Commande fournitures",
  "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@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": {
        "street_number": "37",
        "street": "rue Marcel Philippe",
        "district": "Wallis-et-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": 200,
  "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 :** L'order peut être validé tant que la date d'expiration de la demande d'autorisation n'est pas dépassée. Si cette date est dépassée alors la transaction prend le statut définitif **EXPIRED**, et l'order bascule alors à **FAILED**.

## 11.7. Paiement par vouchers

---

L'API Marketplace permet le paiement partiel ou en intégralité des articles du panier au moyen de vouchers tels que les Titres-Restaurant dématérialisés (TRD), les chèques vacances, etc.

Les moyens de paiement activés pour le moment sont :

Nom d'usage	Contrat acquéreur
Titres-Restaurant	CONECS
Chèque-Vacances Connect	CVCONNECT

### 11.7.1. Prérequis

#### 1. Au niveau marketplace

La marketplace doit être éligible au(x) contrat(s) vouchers envisagé(s).

Pour vous en assurer, effectuez une requête **GET** sur la ressource marketplace à l'attribut "vouchers" :

```
GET /marketplaces/{uuid}
```

Ainsi, pour une marketplace éligible aux contrats **CONECS** et **CVCONNECT**, la réponse contiendra la liste d'objet suivants :

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

Ces paramètres sont renseignés par les services de Lyra Collect. Vous pouvez contacter le support si vous constatez un écart par rapport au paramétrage attendu.

#### 2. Au niveau des sous-marchands

Le numéro de contrat doit être renseigné au niveau du sous-marchand.

Par exemple, pour un sous-marchand n'ayant de contrat actif que sur **CONECS** :

```
GET /sellers/{uuid}
```

Exemple :

```
...
  "vouchers": [
    {
      "contract_type": "CONECS"
      "contract_number": "1999011"
    }
  ]
...
```

Pour créer ou activer un contrat au niveau d'un sous-marchand via l'API d'onboarding, veuillez-vous référer à la [documentation correspondante](#).



## 11.7.2. Création et modification de la commande

Lorsqu'un article est payable par voucher, vous devez l'indiquer en envoyant la liste des contrats concernés au niveau des objets "items", comme suit :

```
"vouchers": [
  {
    "contract_type": "CVCONNECT"
  }
]
```

Dans le cas des contrats **CONECS**, vous devez préciser, en plus, le montant éligible (toujours dans la plus petite unité monétaire de la devise).

Exemple :

```
"vouchers": [
  {
    "contract_type": "CONECS",
    "eligible_amount": 1900
  }
]
```

Vous avez la possibilité de mentionner plusieurs contrats simultanément sur le même article.

Exemple :

```
...
"items": [
  {
    "seller": "6f167596-07f2-4256-9210-7b4ab54fc3b9",
    "reference": "Buffet1",
    "description": "Buffet 1",
    "amount": 8000,
    "type": "FOOD",
    "vouchers": [
      {
        "contract_type": "CVCONNECT"
      },
      {
        "contract_type": "CONECS",
        "eligible_amount": 1400
      }
    ]
  },
  ...
]
...
```

Dans cet exemple, "Buffet 1" est payable :

- par Titre-Restaurant (CONECS) dans la limite de 14 €,
- par Chèque-Vacances (CVCONNECT),
- ou encore par les deux à la fois.

Dans ce cas de figure et en fonction des moyens de paiement choisis par l'acheteur final, la commande peut donc éventuellement conduire à la création d'une, de deux ou de trois transactions.

- 1 seule transaction :
  - par carte classique (CB, VISA, MasterCard, etc.) sur la totalité des 80 €
  - ou par Chèque-Vacances sur 80 €
- 2 transactions :
  - par CONECS + carte classique (CB, VISA, MasterCard, etc.),
  - par CONECS + CVCONNECT,

- ou par CVCONNECT + carte classique (CB, VISA, MasterCard, etc.)
- 3 transactions :
  - par CONECS + CVCONNECT + carte classique (CB, VISA, MasterCard, etc.)

### 11.7.3. Sélection des contrats à l'exécution

Sur chaque commande, un seul marchand peut être réglé par un des types de contrats. Ainsi, si vous indiquez un contrat **CONECS** pour plusieurs sous-marchands, seul l'un d'eux pourra être payé par ce moyen de paiement.

Comment cela fonctionne-t-il? Au moment de l'exécution de la commande, l'API calcule le total par contrat des montants payables par voucher, puis sélectionne, toujours par contrat, le marchand dont le montant est le plus élevé.

Dans l'exemple (simplifié) suivant :

```
...
  "items": [
    {
      "external_ref": "seller1",
      "ref": "Seller 1 Article 1",
      "amount": 2000,
      "voucher": [
        {
          "contract_type": "CONECS",
          "eligible_amount": 900
        }
      ]
    },
    {
      "external_ref": "seller1",
      "ref": "Seller 1 Article 2",
      "amount": 4000,
      "voucher": [
        {
          "contract_type": "CONECS",
          "eligible_amount": 500
        }
      ]
    },
    {
      "external_ref": "seller2",
      "ref": "Seller 2 Article 1",
      "amount": 2000,
      "voucher": [
        {
          "contract_type": "CONECS",
          "eligible_amount": 1900
        }
      ]
    }
  ]
}
```

le **Seller 2** sera sélectionné, car son montant éligible est de 19 € contre 14 € (= 9 + 5) pour le **Seller 1**.

Le principe est le même concernant les contrats **CVCONNECT**.

Et la sélection est indépendante d'un contrat à un autre, c'est-à-dire qu'un sous-marchand peut être sélectionné sur un contrat **CONECS** et un autre sélectionné pour le **CVCONNECT**, alors que les deux proposent simultanément les deux contrats.

#### REMARQUE

Si les montants à départager sont identiques, l'arbitrage s'effectue sur le numéro de contrat.

À l'issue de l'exécution, chaque objet "voucher" des articles concernés reçoit l'attribut `is_selected`, valorisé selon la sélection.

Par exemple :

```
...
```

```

"status":"CREATED",
"amount":8000,
"items":[
{
  "uuid":"fc300ee2-fd43-46e0-9314-770f05a5b338",
  ...
  "reference":"Buffet1",
  "description":"Buffet 1",
  "type":"FOOD",
  "amount":8000,
  ...
  "vouchers":[
  {
    "contract_type":"CVCONNECT",
    "is_selected":true
  },
  {
    "contract_type":"CONECS",
    "eligible_amount":1400,
    "is_selected":true
  }
  ]
}
]

```

#### 11.7.4. Après le paiement

Le choix du moyen de paiement et le montant réellement payé par contrat n'est connu qu'à l'issue du cycle de paiement.

Dans la perspective marketplace, cette information n'est donc disponible qu'à partir du moment où le statut de la commande est passé à **PENDING**.

Si un voucher a été utilisé, l'attribut `actual_amount` renseigne sur le montant réellement affecté à l'article.

Si on reprend l'exemple précédent, nous pouvons donc avoir l'objet "item" suivant :

```

...
"status":"PENDING",
"amount":8000,
"items":[{"
  "uuid":"fc300ee2-fd43-46e0-9314-770f05a5b338",
  ...
  "reference":"Buffet1",
  "description":"Buffet 1",
  "type":"FOOD",
  "amount":8000,
  ...
  "vouchers":[{"
    "contract_type":"CVCONNECT",
    "is_selected":true,
    "actual_amount":3000
  },
  {
    "contract_type":"CONECS",
    "eligible_amount":1400,
    "is_selected":true,
    "actual_amount":1100
  }]
}]

```

On constate dans cet exemple que l'acheteur a utilisé 11€ sur une carte CONECS, plus 30 € sur un Chèque-Vacances. Puisque le montant total est de 80 €, on s'attend donc à avoir une troisième transaction avec le complément (soit  $80 - 30 - 11 = 39$  €).

#### REMARQUE

Étant donné qu'une commande peut contenir plusieurs articles payables par voucher, l'API répartit automatiquement les `actual_amount` dans un ordre arbitraire. Par exemple,

```

...
"items":[{"
  "reference":"Buffet1",
  "description":"Buffet 1",
  "type":"FOOD",
  "amount":8000,
  ...

```

```

"vouchers": [{
  "contract_type": "CONECS",
  "eligible_amount": 1400,
  "is_selected": true,
  "actual_amount": 1400
}]
},
{
"reference": "Buffet2",
"description": "Buffet 2",
"type": "FOOD",
"amount": 5500,
...
"vouchers": [{
  "contract_type": "CONECS",
  "eligible_amount": 1100,
  "is_selected": true,
  "actual_amount": 500
}]
},
{
"reference": "Buffet3",
"description": "Buffet 3",
"type": "FOOD",
"amount": 5500,
...
"vouchers": [{
  "contract_type": "CONECS",
  "eligible_amount": 1100,
  "is_selected": true,
  "actual_amount": 0
}]
}]
...

```

Dans ce cas de figure, l'acheteur semble avoir utilisé le montant de son crédit quotidien de 19 €, qui a été réparti sur le premier article pour la totalité (soit 14 €), sur le second pour une partie (soit 5 €), et rien sur le troisième.

Le crédit de 19 € aurait tout aussi bien pu être réparti de la manière suivante :

- 11 € sur le Buffet 2;
- 8 € sur le Buffet 3;
- 0 € sur le Buffet 1.

### 11.7.5. Modification et annulation

Il est possible d'**annuler** une commande comprenant des transactions payées par des vouchers.

En revanche, la **modification** d'un tel type de transaction n'est pas possible.

## 11.8. Paiement à l'aide d'un lien persistant

---

Vous pouvez générer un lien de paiement persistant pour l'envoyer à l'acheteur final.

Il disposera alors par défaut de 10 jours pour afficher le formulaire de paiement et finaliser sa commande.

### 11.8.1. Générer le lien

Dès la création de l'Order, un lien "persist" est disponible dans l'objet *Order* au niveau de la liste des liens associées "links".

Par exemple :

```
...
  "links": {
    ...
    "persist": {
      "href": "https://secure.lyra.com/marketplace/orders/c4e214db-3ba7-4646-84dc-4e247b1e4b5f/persist"
    }
    ...
  },
  ...
```

Pour générer le lien de paiement, il suffit d'appeler la ressource correspondante :

```
GET /orders/{uuid}/persist
```

**NOTE : Cette ressource effectue une redirection et renvoie la ressource *Order***

Une fois généré, le lien `persist_url` ainsi que la date d'expiration `expiry_date` seront dorénavant accessibles toutes les fois que vous appellerez l'objet *Order* (via un **POST**, un **PUT** ou un **GET**).

Par exemple :

```
...
"persist_url": "https://secure.lyra.com/t/ioeDluRP",
"expiry_date": "2021-02-15T15:37:47.632009Z"
...
```

**NOTE : Dans le cas des prélèvements au fil de l'eau, le montant de la commission est contrôlé dès le premier appel de la ressource `GET /orders/{order}/persist`. Le montant sera à nouveau contrôlé lors de l'appel du formulaire de paiement par l'acheteur final.**

Il n'est possible de générer de liens persistants que sur les *Order* au statut "**CREATED**".

Il n'est pas possible de modifier ni de supprimer le lien.

### 11.8.2. Adapter le délai d'expiration

Par défaut, le délai d'expiration du lien est de 10 jours.

Pour modifier ce délai, ajoutez le paramètre "*delay*" lors de la requête de génération du lien.

Par exemple :

```
GET https://secure.lyra.com/marketplace/orders/3c58e343-abbd-4c2b-9cac-42db158987b0/persist?delay=5
```

Il n'est pas possible de modifier le délai d'expiration après sa création.

## 12. ANALYSER LE RÉSULTAT DU PAIEMENT

Le serveur de la Marketplace est notifié sous forme de *webhook* pour chaque changement de statut des objets Order et Token.

Le *webhook* est envoyé à l'adresse renseignée par la marketplace lors de la phase d'intégration.

Cette URL est lisible sur l'objet Marketplace à l'attribut "**webhook\_url**".

Le *webhook* est envoyé en POST avec l'entête "**Content-Type: application/json**".

### Exemple de notification (webhook)

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

En PHP, ce Json peut être récupéré par le code suivant :

```
$json = file_get_contents('php://input');  
$data = json_decode($json);  
$orderId = $data->order ?? false;
```

Il appartient ensuite à la Marketplace d'appeler la ressource GET ORDER (ou TOKEN) avec l'UUID communiqué afin de connaître le nouvel état de l'objet en question.

```
GET /orders/{uuid}
```

### Exemple d'appel GET ORDER

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

### Exemple de retour GET ORDER

```
{  
  "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": "Commande",  
  "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": 200,
"url_return": "http://www.lyra-sms.com/",
"links": {
    "items": {
        "href": "https://secure.lyra.com/marketplace/orders/515abac9-6cb2-4e21-8a25-b08d7e41e43c/items"
    },
    "transactions": {
        "href": "https://secure.lyra.com/marketplace/orders/515abac9-6cb2-4e21-8a25-b08d7e41e43c/transactions"
    },
    "refunds": {
        "href": "https://secure.lyra.com/marketplace/orders/515abac9-6cb2-4e21-8a25-b08d7e41e43c/refunds"
    },
    "execute": {
        "href": "https://secure.lyra.com/marketplace/orders/515abac9-6cb2-4e21-8a25-b08d7e41e43c/execute"
    },
    "execute-embedded": {
        "href": "https://secure.lyra.com/marketplace/orders/515abac9-6cb2-4e21-8a25-b08d7e41e43c/execute-embedded"
    },
    ...
},
"vads_transaction_id": 500003,
"vads_transaction_date": ""
}

```

En cas de doute sur l'envoi ou la réception du *webhook*, vous pouvez consulter l'attribut "**webhook\_result**" de la ressource en question. Cet attribut contient le "status code" de la requête HTTP réalisée par notre serveur. Ainsi,

- s'il est valorisé à null, le webhook n'a pas (encore) été envoyé.
- S'il indique 200, le webhook est réputé vous être parvenu.
- S'il indique 404, nous vous invitons à contrôler le webhook que vous avez déclaré.
- etc...

Remarque :

Le code d'autorisation de la transaction est enregistré dans l'attribut **auto\_code** renvoyé par le détail des transactions (accessible par la ressource GET Transaction (voir [Objet Transaction](#) à la page 80).

## 13. GESTION DES ALIAS

---

L'API Marketplace utilise deux notions :

- les **alias**
- les **tokens**

Un **alias** est un objet qui permet de stocker les données d'un moyen de paiement.

Un **token** est un objet qui est associé à un alias. Il contient notamment les informations de l'acheteur.

Il y a 2 façons de créer un alias (et donc un token) :

1. durant le paiement (voir chapitre [Paiement avec création d'un alias](#) à la page 33). L'alias et le token sont donc associés à un Order,
2. sans passer par la création d'un Order (voir chapitre suivant).

### 13.1. Créer un token

---

Il est possible de demander l'enregistrement du moyen de paiement (et donc la création d'un token) sans qu'il ne soit lié à une commande.

Pour cela, il faut appeler l'API TOKEN pour créer un token, avec une requête proche de celle utilisée pour une création de paiement.

#### Créer un token en utilisant la page de paiement

```
POST /tokens/
```

Exemple :

Requête

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

Corps

```
{
  "marketplace": "9537e049-8862-400a-ae8d-da2ec9ca6051",
  "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": "Acheteur_1",
    "address": {
      "zipcode": "59259",
      "street_number": "37",
      "country": "FR",
      "street": "rue Marcel Philippe",
      "city": "Vignoux"
    }
  }
}
```

Réponse

```
{
```



```
"token": "d3329266-c8d6-421c-8d2a-10a8ffbcaef6",
"payment_url": "https://secure.lyra.com/vads-payment/
exec.refresh.a;jsessionid=335D2aDb5eF8356Aed2cf3dF.vadpayment02inte01lbg?
cacheId=913355311811276000040"
}
```

Une fois le token créé, il faut rediriger l'acheteur vers l'URL transmise dans **payment\_url** pour procéder à l'enregistrement du moyen de paiement (et donc à la création d'alias).

### Créer un token en utilisant le formulaire embarqué

```
POST /tokens/embedded/
```

Exemple :

Requête

```
POST https://secure.lyra.com/marketplace/tokens/embedded/
```

Corps

```
{
  "marketplace": "9537e049-8862-400a-ae8d-da2ec9ca6051",
  "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": "Acheteur_1",
    "address": {
      "zipcode": "59259",
      "street_number": "37",
      "country": "FR",
      "street": "rue Marcel Philippe",
      "city": "Vignoux"
    }
  }
}
```

Réponse

```
{
  "token": "d3329266-c8d6-421c-8d2a-10a8ffbcaef6",
  "form_token": "fa0yfV2FQuR3aak1SwsgcuZg195eyJhbW91bnQiOjEwOTAwLCJjdXJyZW5jeSI6IkVVUiiIsIm
1vZGUioiJURVNUiwiidmVyc2lvbiI6Mywib3JkZXJJZCI6IjVjNDk1N2E5LWNkOGEtNGJhZC1hMGNiLTgxMT
ZiMWE3ZDdmNiIsInNob3BOYW11IjoithlyYSBTTVMiLCJicmFuZFByaW9yaXR5IjpbIkJBTkNPTlRBQ1QiL
CUDQiIsIkUtQ0FSVEVCEVVRISik1BU1RFUkNBkQiLCVGVVVMVCJ9fX0a702"
}
```

Une fois le token créé, le **form\_token** doit être inséré dans le formulaire (attribut **kr-form-token**) pour procéder à l'enregistrement du moyen de paiement (et donc à la création d'alias).

## 13.2. Analyser le résultat d'une demande de token

---

Comme pour un paiement, la Marketplace sera notifiée par [webhook](#) de l'évolution d'une demande de token.

Exemple de notification (webhook):

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

L'identifiant ainsi transmis permettra à la Marketplace de faire un GET TOKEN afin de savoir si la création de l'alias a réussi.

```
GET /tokens/{uuid}
```

**Remarque :**

Lors d'un paiement avec création d'alias, vous recevrez donc deux notifications : une pour la création de l'Order et une pour la création de l'alias (il faudra utiliser l'ID du token retourné dans la notification pour retrouver l'alias correspondant).

Exemple d'appel GET TOKEN :

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

Exemple de retour GET TOKEN :

```
{
  "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": "Acheteur_1",
    "title": "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": "SUCCEEDED",
  "alias": "1144951ea7ab42989c97159b3dfc0382",
  "alias_to_update": null,
  "payment_url": "https://secure.lyra.com/vads-payment/exec.refresh.a;jsessionid=Cdb37C1CcfC5eA2BE82bDCA6?cacheId=913355311811206000040"
}
```

## 13.3. Modifier un alias

---

Pour mettre à jour un moyen de paiement, vous devez procéder à une nouvelle création de token en ajoutant l'alias existant dans l'attribut **alias\_to\_update** de la requête POST TOKEN.

### **Mettre à jour un alias en utilisant la page de paiement**

Exemple :

Requête

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

Corps

```
{
  "marketplace": "9537e049-8862-400a-ae8d-da2ec9ca6051",
  "alias_to_update": "1144951ea7ab42989c97159b3dfc0382",
  "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": "Acheteur_1",
    "address": {
      "zipcode": "59259",
      "street_number": "37",
      "country": "FR",
      "street": "rue Marcel Philippe",
      "city": "Vignoux"
    }
  }
}
```

Réponse

```
{
  "token": "62a52e50-ce29-409a-9cec-9ea6ee36ab41",
  "payment_url": "https://secure.lyra.com/vads-payment/
exec.refresh.a;jsessionid=335D2aDb5eF8356Aed2cf3dF.vadpayment02inte01lbg?
cacheId=913355311811276000040"
}
```

Pour procéder à la mise à jour du moyen de paiement, il faut rediriger l'acheteur vers l'URL transmise dans **payment\_url**.

Vous recevrez une notification webhook à la fin de l'opération.

### **Mettre à jour un alias en utilisant le formulaire embarqué**

Exemple :

Requête

```
POST https://secure.lyra.com/marketplace/tokens/embedded/
```

Corps

```
{
  "marketplace": "9537e049-8862-400a-ae8d-da2ec9ca6051",
  "alias_to_update": "1144951ea7ab42989c97159b3dfc0382",
  "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": "Acheteur_1",
"address": {
  "zipcode": "59259",
  "street_number": "37",
  "country": "FR",
  "street": "rue Marcel Philippe",
  "city": "Vignoux"
}
}
```

## Réponse

```
{
  "token": "62a52e50-ce29-409a-9cec-9ea6ee36ab41",
  "form_token": "fa0yfv2FQuR3aak1SwsgcuZg195eyJhbW91bnQiOjEwOTAwLCJjdXJyZW5jeSI6IkVVUuIIsIm
1vZGUuOiJURVNUiwiidmVyc2lvbiI6Mywib3JkZXJJZCI6IjVjNDk1N2E5LWNkOGEtNGJhZC1hMGNiLTgxMT
ZiMWE3ZDdmNiIsInNob3BOYWllIjoIiThlyYSBTTVMiLCJicmFuZFByaW9yaXR5IjpbIkJBTkNPTlRBQ1QiL
CjDQiIsIkUtQ0FSVEVCTEVVRSIsIk1BU1RFUkNBUCkqIjE5LWVhZC1hMGNiLTgxMT"
}
```

Pour procéder à la mise à jour du moyen de paiement, le **form\_token** doit être inséré dans le formulaire (attribut **kr-form-token**).

## 13.4. Récupérer le détail d'un alias

---

Une fois qu'un alias a été créé, il est possible d'obtenir ses informations en appelant le service de détail d'un token.

```
GET /marketplaces/{marketplace}/alias/{alias}
```

Exemple :

Requête

```
GET https://secure.lyra.com/marketplace/marketplaces/9537e049-8862-400a-ae8d-da2ec9ca6051/alias/1144951ea7ab42989c97159b3dfc0382
```

Réponse

```
{
  "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"
}
```

Autre exemple de réponse avec un alias de type IBAN :

```
{
  "brand": "SDD",
  "expiry_month": "5",
  "expiry_year": "2023",
  "number": "FR7630002005701234567890158_CRLYFRPPXXX",
  "first_name": "Jean",
  "last_name": "Dupond",
  "email": "jean.dupond@lyra.fr",
  "creation_date": "2020-05-12T14:24:13Z"
}
```

## 14. TRAITER LE RETOUR VERS LA MARKETPLACE

---

Afin de rediriger l'acheteur vers la Marketplace, plusieurs attributs de la ressource ORDER peuvent être renseignés :

- **"url\_return"** est l'URL de retour par défaut. Vous pouvez n'utiliser que cet attribut si vous ne souhaitez pas différencier les cas de retour. Les attributs qui suivent surchargent sa valeur au cas par cas.
- **"url\_success"** est l'URL appelée si le paiement est un succès.
- **"url\_refused"** est l'URL appelée si le paiement est refusé.
- **"url\_cancel"** est l'URL appelée si le paiement est annulé.
- **"url\_error"** est l'URL appelée si le paiement est en erreur.

Par défaut, les données du paiement sont envoyées à l'URL de retour sous la forme d'un formulaire HTTP GET (dans la « query string »).

Exemple :

```
https://mymarketplace.com/return_to_shop?ref=1234&customer=ABCD
```

Ce comportement peut être surchargé via l'attribut **"return\_mode"** qui peut prendre les valeurs suivantes :

- **'NONE'** : aucun paramètre n'est envoyé à l'URL de retour.
- **'POST'** : les paramètres sont envoyés à l'URL de retour sous la forme d'un formulaire HTTP POST (Si le retour boutique se fait sur un environnement non https alors le navigateur affichera un pop-up de sécurité à l'acheteur.)
- **'GET'** (par défaut) : Les champs de retour sont transmis à l'URL de retour sous la forme d'un formulaire HTTP GET (dans la « query string »).

**Remarque** : le retour vers la Marketplace doit uniquement vous permettre d'afficher un contexte visuel à l'acheteur. N'utilisez pas les données reçues pour effectuer le traitement en base de données, ni pour contrôler le statut du paiement.

## 15. MODIFIER UNE COMMANDE

Il est possible de modifier un Order tant qu'il possède un statut transitoire (**CREATED** ou **PENDING**).

La transaction initiale ne doit donc pas être remise en banque.

Dans le cas d'un Order au statut **CREATED**, toutes les valeurs sont modifiables, sans restriction (voir l'étape 2 du chapitre *Réaliser un paiement* à la page 25).

Dans le cas d'un Order au statut **PENDING**, la commande PUT est rejetée si les deux valeurs suivantes sont identiques à celles du dernier enregistrement :

- **amount**, déterminé à partir de la somme des montants des items.  
La requête de modification annule et remplace l'ensemble des items de l'Order initial.
- **expected\_capture\_date** en UTC au format YYYYMMDDHHMMSS.  
La comparaison des `expected_capture_date` ne tient compte que de la date.  
Par exemple, 20200101100000 (1er janvier 2020 à 10h UTC) est équivalent à 20200101180000 (1er janvier 2020 à 18h UTC).

Par ailleurs

- le montant ne peut être supérieur au montant initial de la commande (cf. valeur enregistrée à la clé **initial\_amount**),
- les items doivent être obligatoirement transmis dans la requête PUT (?expand=items).

Exemple d'appel PUT ORDER :

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

```
{
  "marketplace": "ebfb36ab-2d30-4326-adb9-e16b0c9a89f3",
  "reference": "Exemple MKP BURO",
  "description": "Commande fournitures",
  "currency": "EUR",
  "url_return": "https://URLreturn.com",
  "expected_capture_date": "202006205352",
  "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": "Stylos verts",
      "amount": 1000,
      "is_commission": false
    },
    {
      "seller": "d0f80202-0676-4d8d-9247-f455f30aec1b",
      "reference": "liv",
      "description": "Livraison",
      "amount": 500,
      "is_commission": false
    }
  ],
  "buyer": {
    "type": "PRIVATE",
    "email": "jean.dupont@acheteur.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"
}
}
```

## 15.1. Mise à jour d'une commande réglée en plusieurs fois

---

Il est possible de modifier les transactions après le paiement et tant que les transactions en question sont en attente (c'est-à-dire qu'elles doivent se trouver à l'état PENDING).

Seules deux valeurs sont modifiables :

- le montant de la transaction (attribut **amount**) ;
- la date de capture de la transaction (attribut **expected\_capture\_date**).

Notes :

- Tout comme avec une commande mono-transactionnelle, il n'est pas possible d'accroître le montant d'une transaction au-delà de son montant initial, tel qu'enregistré dans l'attribut en lecture seule **initial\_amount**.
- Les items doivent être obligatoirement transmis dans la requête PUT (?expand=items).
- Il est possible de supprimer une (ou plusieurs) transaction(s) (cf. ci-dessous), mais il n'est actuellement pas possible d'en ajouter.

### Procédure :

La modification d'une ou de plusieurs transactions passe par la même commande de mise à jour d'une commande (PUT ORDER), mais avec deux différences :

1. Il faut ajouter la liste complète des transactions de la commande dans le corps de la requête PUT ORDER dans une liste de **transactions**, comprenant chacune les attributs **trans\_uid**, **expected\_capture\_date** et **amount** éventuellement modifiés.

Par exemple :

```
{
  "marketplace": "2434c0a2-9d46-4e96-9553-1536c898625b",
  "reference": "MyMultitransactionOrder01",
  "description": "Mise à jour de la commande.",
  "currency": "EUR",
  "url_return": "https://www.my-website.com/",
  "language": "en",
  "items": [
    {
      "seller": "4d20a9d4-0526-4474-b452-e936dc25418d",
      "reference": "0000001",
      "description": "TV Ecran plat",
      "amount": 390000,
      "quantity": 1,
      "is_commission": false
    },
    {
      "seller": "72ccc2ff-b455-4653-847e-deb6fee99f8d",
      "reference": "0000002",
      "description": "Commission",
      "amount": 14755,
      "quantity": 1,
      "is_commission": true
    }
  ],
  "transactions": [
    {
      "trans_uid": "c383739d12a6489badc3bb6847db84cc",
      "payment_scheme": "CB",

```



```

    "amount": 14755,
    "expected_capture_date": "201909185743"
  },
  {
    "trans_uuid": "d0c4d34249d540af87ff8df3a2fa314a",
    "payment_scheme": "CB",
    "amount": 190000,
    "expected_capture_date": "201905175352"
  },
  {
    "trans_uuid": "8c5a6788b3334d368185b0a567dd7bcd",
    "payment_scheme": "CB",
    "amount": 200000,
    "expected_capture_date": "201904205352"
  }
],
"buyer": {
  ...
}
}

```

## 2. Il est impératif d'ajouter "transactions" dans l'attribut url « expand ».

Vous devez donc envoyer une ressource PUT ORDER sur

**[https://secure.lyra.com/marketplace/orders/{order\\_uuid}?expand=items,transactions](https://secure.lyra.com/marketplace/orders/{order_uuid}?expand=items,transactions)**.

Si la valeur "transactions" n'est pas indiquée, aucune modification de transaction ne pourra être prise en compte.

Lors de la mise à jour, l'application vérifie la cohérence des montants entre les items et les transactions : en toute logique, le total des montants de l'item doit être identique au montant total des transactions. Et par conséquent, une réduction des montants sur les items doit mécaniquement s'accompagner d'une réduction des montants sur les transactions.

### *Cas particulier d'une suppression de transactions*

Il se peut que la réduction des montants sur la partie items soit supérieure au montant d'une des transactions. Si au lieu de répartir la réduction sur l'ensemble des transactions, vous choisissez de supprimer une transaction, il suffit de l'omettre dans la liste des transactions envoyées en requête de modification. Elle est alors interprétée comme annulée, et son statut passe à CANCELLED.

Autres remarques :

- L'application renvoie une erreur si la mise à jour ne comporte aucune modification (ni des transactions, ni des items, ni des valeurs de la commande).
- La nouvelle valeur de **expected\_capture\_date** ne doit pas être antérieure à la date et heure de modification.

## 16. ANNULER UNE COMMANDE

---

L'annulation de la commande est possible sur les commandes avant paiement et après paiement tant que ce dernier n'a pas été remisé en banque, c'est à dire lorsqu'elle se trouve à l'état:

- **CREATED** et *avant l'exécution de la commande*,
- **PENDING**.

Si vous essayez d'annuler le paiement entre le moment de l'exécution de la commande (alors encore à l'état CREATED) et la fin du processus du paiement (i.e. le passage de CREATED à PENDING), vous recevrez l'erreur suivante :

```
{'Order': "This order has been executed and is not cancelable until the end of the payment process"}
```

Pour annuler une commande vous devez faire un DELETE sur la ressource ORDER.

```
DELETE /orders/{uuid}
```

Exemple :

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

En cas de succès, le serveur répond par un code **HTTP 204** (NO CONTENT).

## 17. REMBOURSER UN PAIEMENT

---

Pour rembourser un paiement, vous devez indiquer sur quelle commande (ORDER) porte le remboursement et quel vendeur (SELLER) va assumer le coût du remboursement.

Pour demander le remboursement d'un paiement, vous suivez les 2 étapes suivantes :

1. Créer une demande de remboursement
2. Suivre l'évolution de la demande

### 17.1. Créer la demande de remboursement

---

Cette étape permet de créer la demande de remboursement en faisant un POST sur la ressource REFUND :

```
POST /refunds/
```

A la création d'une demande de remboursement, les éléments suivants sont contrôlés par l'API et font l'objet d'un code de retour d'erreur 400 s'ils ne sont pas vérifiés :

- La commande à rembourser doit être dans l'état SUCCEEDED.
- Le vendeur qui rembourse doit être un des vendeurs de la commande remboursée.
- Le remboursement ne doit cibler qu'un seul vendeur. Si plusieurs vendeurs doivent rembourser une partie d'une même commande, plusieurs demandes de remboursements distinctes devront être créées : une pour chaque vendeur. La seule exception étant si le deuxième vendeur de la demande est le gestionnaire de la place de marché.

**Remarque** : le montant remboursé ne fait pas l'objet de contrôle à cette étape. Le contrôle a lieu ultérieurement, par les chaînes de traitement planifiées, ce qui se traduit par l'évolution du statut de la demande de traitement.

## Exemple d'appel POST REFUND

Dans cet exemple de contenu JSON à poster, une demande de remboursement de 130€ est formulée, dont 120€ à la charge du vendeur et 10€ à la charge du gestionnaire de la place de marché:

### Requête

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

### Corps

```
{
  "order": "9537e049-8862-400a-ae8d-da2ec9ca6051",
  "reference": "remb000045",
  "description": "Remboursement Mme Lafont 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": "Gestionnaire",
      "amount": 1000
    }
  ]
}
```

## Exemple réponse du POST REFUND

L'attribut **amount** en retour est calculé automatiquement à partir des montants postés de chacune des 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": "Remboursement Mme Lafont 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": "Gestionnaire",
      "amount": 1000,
      "is_commission": false
    }
  ]
}
```

## 17.2. 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, enregistrez un [webhook](#) sur l'objet refund et/ou consultez le statut du REFUND par un GET :

```
GET /refunds/{uuid}
```

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 :

Requête

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

Réponse

```
{
  "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
    }
  ]
}
```

## 17.3. Modifier une demande de remboursement

Un remboursement en cours peut être modifié dans la limite de son montant initial tant qu'il n'a pas été exécuté (statut CREATED ou PENDING).

La modification s'effectue à l'aide de l'uuid du remboursement en question, à travers la ressource:

```
PUT /refunds/{uuid}
```

**i** **Remarque :** Si le montant du remboursement est inchangé, le serveur renvoie une erreur **HTTP 400**.

Exemple :

Requête

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

Corps

```
{
  "order": "9537e049-8862-400a-ae8d-da2ec9ca6051",
  "reference": "remb000045",
  "description": "Remboursement Mme Lafont 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": "Gestionnaire",
      "amount": 900,
      "is_commission": false
    }
  ]
}
```

Réponse

```
{
  "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": "Remboursement Mme Lafont 001",
  "amount": 10905,
  "currency": "EUR",
  "status": "PENDING",
  "transaction": null,
  "items": [
    {
      "seller": "dfc42a76-10b5-421a-91cd-c288c8265c92",
      "refund": "83402cc0-d969-443b-a72a-0f2fe9557879",
      "item": "9f7c6d31-deff-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": "Gestionnaire",
"amount": 900,
"is_commission": false
}
]
```

## 17.4. Annuler une demande de remboursement

---

Un remboursement en cours peut être annulé tant qu'il n'a pas été exécuté (statut CREATED ou PENDING).

```
DELETE /refunds/{uuid}
```

Par exemple :

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

En cas de succès, le serveur répond par un code **HTTP 204** (NO CONTENT).

Si le statut du remboursement ne permet pas l'annulation, le serveur retourne une erreur **HTTP 405**.

## 18. DÉCLENCHEMENT MANUEL DU PAIEMENT DES ITEMS

Dans la cinématique normale, le paiement d'un item de commission intervient automatiquement après l'expiration du délai de blocage des fonds du vendeur lié à l'item.

Il peut arriver que la marketplace souhaite retenir ce paiement afin de contrôler le moment de son déclenchement.

C'est notamment le cas d'une prestation qui s'étale dans le temps. Dans ce cas le client a effectué son paiement avant le début de la prestation, mais le vendeur ne doit recevoir sa contrepartie que suite à la vérification par la marketplace du bon déroulement de la prestation.

Il appartient alors à la marketplace :

- d'indiquer lors de la création de la commande le ou les items concernés par le blocage du paiement, en valorisant l'attribut **hold\_payment** à **true**:

```
"items": [
  {
    "seller": "4d20a9d4-0526-4474-b452-e936dc25418d",
    "reference": "croisiere12345",
    "description": "Croisiere en bateau",
    "amount": 245000,
    "quantity": 1,
    "is_commission": false,
    "hold_payment": true
  },
  {
    "seller": "4d20a9d4-0526-4474-b452-e936dc25418d",
    "reference": "cartemaritime",
    "description": "Carte maritime - La Martinique",
    "amount": 1250, "quantity": 1,
    "is_commission": false
  },
  {
    "seller": "72ccc2ff-b455-4653-847e-deb6fee99f8d",
    "reference": "croisiere12345com",
    "description": "Commission sur croisiere en bateau",
    "amount": 16450,
    "quantity": 1,
    "is_commission": true
  }
]
```

- et, au moment voulu, de libérer le transfert en effectuant une requête POST (sans corps de requête) à l'adresse :

```
POST /items/{uuid}/activate
```

En cas de succès de l'opération, la requête est redirigée (status\_code = 302) vers le détail de l'item, qui indique alors **hold\_payment** : **false**.



### Remarque :

- Le blocage ne peut concerner les items de panier d'un vendeur de type Marketplace (seller avec "is\_marketPlace = true").
- Une fois le paiement de l'item activé, il n'est plus possible de le bloquer à nouveau .
- L'attribut **hold\_payment** n'est pas obligatoire. Il est même préférable de l'omettre (ou bien de le valoriser à "**hold\_payment** : **null**") pour tout item dont le transfert doit être géré automatiquement, afin de distinguer les items ayant fait l'objet d'un blocage des autres.



## 19. CONSULTER LES CASHOUTS DE LA MARKETPLACE

### 19.1. Comprendre les cashouts

Le cashout représente un virement du compte de cantonnement du sous-marchand vers son propre compte bancaire.

L'opération est réalisée automatiquement à **J + d** jours après le paiement, sauf si l'opérateur Marketplace l'a bloquée pour *l'activer manuellement*.

- *J* étant le *décal de capture de la transaction*.
- *d* étant le "décal de cashout" du sous-vendeur tel que paramétré par l'opérateur Marketplace (voir l'étape **Créer le vendeur** dans *Marketing-Onboarding KYC*).

Par exemple, M. Martin paie une commande aujourd'hui (*J*). Si la remise s'effectue demain (*J+1*) et si la Marketplace a défini un décal de cashout de 2 jours sur le sous-marchand, le virement interviendra dans 3 jours.

Son montant correspond donc à la somme des transactions réalisées (et remises) *d* jours auparavant moins les remboursements effectués (et remisés) entre temps.

Par exemple, si le décal de cashout est de 2 jours et que 4 transactions de 250€ ont été remises avant-hier, dont un article remboursé hier, le montant du cashout de ce jour sera égal à :  $(4 - 1) \times 250 \text{ €} = 750 \text{ €}$ .

### 19.2. Cinématique du cashout

L'objet `transfer` fait la médiation entre la transaction et le cashout.

En effet, dans le cadre d'une Marketplace :

- une `transaction` peut concerner plusieurs articles de plusieurs sous-vendeurs différents, pour un seul *acheteur*
- et un `cashout` agrège plusieurs `transactions` de plusieurs commandes différentes, pour un seul *vendeur*.

Le `transfer` représente la part qu'un article occupe pour une seule transaction (sachant qu'une commande peut elle-même, être répartie en plusieurs transactions).

Il est entièrement géré par l'API Marketplace.

Par exemple, une Marketplace donne à M. Martin la possibilité de payer ses 3 articles ci-dessous en 3 fois. Les transferts représentent le corps du tableau, tandis que les marges en ligne représentent les articles, et les marges en colonne correspondent aux transactions.

		Transactions			Total Articles
		1ère tranche	2ème tranche	3ème tranche	
Articles	Ordinateur	740,88 €	740,88 €	518,23 €	1 999,99 €
	Grand écran	926,10 €	926,10 €	647,79 €	2 499,99 €
	Barre de son	333,02 €	333,02 €	232,96 €	899,00 €
Total Transactions		2 000,00 €	2 000,00 €	1 398,98 €	5 398,98 €

Dans cette commande, 9 transferts sont donc générés, en utilisant une règle de répartition proportionnelle. Par exemple, le transfert correspondant au grand écran sur la première transaction est de **926,10€ (= 2000,00 x 2499,99 / 5398,98)**. Et si vous faites les calculs exacts, vous pourrez remarquer qu'un article (ici, *la barre de son*) et qu'une transaction (*la dernière*) sont utilisés pour rectifier l'arrondi au centime près.

Dans la cinématique globale de la Marketplace, le `transfer` est en quelques sortes *la plus petite unité de décomposition d'une commande*, et la clé de transformation des *cash-in en cash-out*. Étant à l'articulation des deux mouvements, il est généré dès que la transaction est identifiée comme étant remise (c'est-à-dire passée au statut **SUCCEEDED**). Il prend alors le statut **CREATED**.

Une fois le cash-out remisé, celui-ci passe au statut **SUCCEEDED**. Ce statut est communiqué au `transfer` qui lui est associé, et qui passe lui aussi au statut **SUCCEEDED**.

#### REMARQUE

Par définition, les produits vendus en propre par l'opérateur de la Marketplace ne font pas l'objet de cashouts. Donc *seuls les sous-marchands de la Marketplace sont concernés par les virements*. Et par conséquent, les transferts en direction de la Marketplace (vente en propre et `item commission`) conservent le statut **CREATED** pendant toute leur durée de vie.

## 19.3. Identifier le cashout et les commandes associées

Comment établir la connexion entre le cashout d'un seller à une commande réalisée sur la Marketplace, à partir des informations renvoyées par le Back Office Expert et celles de l'API Marketplace ?

Dans le Back Office Marchand et les relevés de transactions, les cashouts sont identifiables à travers leurs `capture_label`.

Par exemple, `LC_22372582900407` pour "Lyra Collect", ou autre exemple, une association entre le nom de la Marketplace et un numéro de virement, comme `MYMKPSLR88888444`.

En repérant le cashout concerné dans la [liste des cashouts](#), vous pouvez ensuite interroger le [détail du cashout](#), et ainsi retrouver les commandes, les items et les remboursements associés.

## 19.4. Liste des cashouts

La liste des cashouts d'une marketplace est consultable sur l'URL :

```
GET /cashouts/
```

Les résultats sont retournés sous la forme:

```
{
  "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": "MYMKPSLR1012324"
    },
    {
      "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": "MYMKPSLR1012325"
    },
    {
      "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": "MYMKPSLR1012326"
}
]
```

Ces résultats sont triés par date de capture descendante (attribut **captured\_at**).

Ils sont affichés de façon paginée. Il y a 100 cashouts par page.

Les liens "**next**" et "**previous**", quand ils sont valorisés, permettent de naviguer d'une page à une autre.

Il est également possible de filtrer les résultats de cette liste par leur date de capture.

Pour ce faire, il faut spécifier les paramètres **capture\_start\_date** et/ou **capture\_end\_date** de la façon suivante :

```
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
```

Si le paramètre **capture\_start\_date** est renseigné mais pas **capture\_end\_date**, celui-ci est fixé à la date du jour (« aujourd'hui »).

Si à l'inverse, c'est le paramètre **capture\_start\_date** qui est omis mais que **capture\_end\_date** est précisé, la sélection prendra les cashouts depuis le début jusqu'à la date donnée.



**Remarque :** Les cashouts dont l'attribut "captured\_at" n'est pas valorisé sont exclus des résultats.

## 19.5. Détails des cashouts

---

La liste des cashouts fournit l'URL pour accéder aux détails de chacun des cashouts, par exemple:

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

Un GET sur l'URL permet de récupérer les informations sur les virements et les remboursements associés au cashout (avec les détails des commandes et items) :

```
GET /cachouts/{uuid}
```

Exemple :

Requête

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

Réponse

```
{
  "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": "Repas"
      }
    },
    {
      "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": "Repas"
      }
    },
    {
      "amount": 450,
      "currency": "EUR"
    }
  ]
}
```

## 20. GÉNÉRER UN CLIENT PAR OPENAPI

---

L'API Marketplace exploite **Swagger** (<https://swagger.io/>) pour faciliter la génération d'un client prêt à l'emploi, et ce dans une grande variété de langages.

Comment procéder?

1. Récupérez le modèle de l'API Marketplace depuis cette adresse :

- au format yaml: [https://secure.lyra.com/marketplace/open\\_api.yaml](https://secure.lyra.com/marketplace/open_api.yaml)
- au format json: [https://secure.lyra.com/marketplace/open\\_api.json](https://secure.lyra.com/marketplace/open_api.json)

2. Copiez le modèle dans la fenêtre gauche de l'éditeur Swagger (<https://editor.swagger.io/>).

Vous pouvez ignorer sans danger l'erreur "Semantic error at paths./orders/.post.operationId Operations must have unique operationIds."

3. Dans le menu de l'éditeur Swagger, cliquez sur **Generate Client** et sélectionnez le langage de votre choix (PHP, Python, etc.).

4. Après téléchargement et extraction du code généré, suivez les instructions du fichier README.md.

Et voilà!

## 21. DICTIONNAIRE DE DONNÉES

### 21.1. Objet Address

Tous les attributs de cet objet sont obligatoires si au minimum un attribut est renseigné. Donc si aucun attribut n'est renseigné, aucun attribut n'est obligatoire.

Nom	Description	Format	Obligatoire	Exemple
street_number	Numéro de la rue	an..5		12
street	Nom de la rue	ans..255		rue de la bohème
district	Complément d'adresse	ans..127		quartier 15ème
zipcode	Code postal	an..64		75015
city	Ville	an.128		Paris
state	Etat	ans..127		Paris
country	Pays	a2		FR pour France GP pour Guadeloupe PF pour Polynésie Française

### 21.2. Objet Alias

Nom	Description	Format	Exemple
brand	Marque de la carte	string	CB
expiry_month	Mois d'expiration de la carte entre 1 et 12	n..2	3
expiry_year	Année d'expiration sur 4 chiffres	n4	2023
number	Numéro de carte masqué	string	497010XXXXXX0000
first_name	Prénom de l'acheteur	string	Jean
last_name	Nom de l'acheteur	string	Dupond
email	E-mail de l'acheteur	string	jean.dupond@lyra.com
creation_date	Date de création de l'alias	cf. exemple	2020-04-21T16:01:19Z

## 21.3. Objet Buyer

Nom	Description	Format	Obligatoire	Exemple
reference	référence de l'acheteur pour la Marketplace	an..63	X	000012
legal_name	nom de la société	an..63		Dupond & Cie
title	civilité de l'acheteur	an..63		Mme
type	type d'acheteur	enum	X	PRIVATE   COMPANY
first_name	prénom de l'acheteur	an..63		Stéphanie
last_name	nom de l'acheteur	an..63		DUPOND
phone_number	numéro de téléphone de l'acheteur	an..32	X	0612324565
email	adresse e-mail de l'acheteur	ans..150	X	ste.dupond@site.com
address	adresse postale de l'acheteur	objet Address		

## 21.4. Objet Item

Nom	Description	Format	Obligatoire	Exemple
uuid	Identifiant unique de la prestation <i>Généré par Lyra Collect</i>	an..36	N/A	3ea5e574-e198-55d4-ba23-f9405ec4226f
href	URL d'accès à cette ressource <i>Généré par Lyra Collect</i>	cf. exemple	N/A	<a href="https://secure.lyra.com/marketplace/items/3ea5e574-e198-55d4-ba23-f9405ec4226f">https://secure.lyra.com/marketplace/items/3ea5e574-e198-55d4-ba23-f9405ec4226f</a>
created_at	Date et heure de création de la ressource <i>Généré par Lyra Collect</i>	cf. exemple	N/A	2015-01-17T09:39:54.948Z
updated_at	Date et heure de la dernière modification de la ressource <i>Généré par Lyra Collect</i>	cf. exemple	N/A	2015-01-17T09:39:54.948Z
seller	UUID du prestataire	an..36	X	1ea5e574-e198-55d4-ba23-f9405ec4226a
order	UUID de la commande	an..36		8ea5e574-e198-55d4-ba23-f9405ec4226c
reference	Référence de la prestation	an..32+"-"	X	0000000181
description	Description de la prestation	an..255		Prestation blablabla
type	définit le type de l'article contenu dans le panier	enum		voir tableau ci-après.
amount	Si <b>is_commission</b> est valorisé à false : Montant de la prestation. Si <b>is_commission</b> est valorisé à true : Montant de la commission définie au niveau de la commande.	n..12	X	50000
commission_amount	Montant de la commission définie sur l'item.	n..12		500

Nom	Description	Format	Obligatoire	Exemple
is_commission	indique que cet item est une commission	enum		<i>true / false</i> <i>false</i> par défaut
status	Statut de la prestation <i>Généré par Lyra Collect</i>	an..10	N/A	<i>SUCCEDED</i>

### Valeurs de l'attribut type

Valeur	Description
FOOD_AND_GROCERY	Produits alimentaires et d'épicerie
AUTOMOTIVE	Automobile / Moto
ENTERTAINMENT	Divertissement / Culture
HOME_AND_GARDEN	Maison et jardin
HOME_APPLIANCE	Equipement de la maison
AUCTION_AND_GROUP_BUYING	Ventes aux enchères et achats groupés
FLOWERS_AND_GIFTS	Fleurs et cadeaux
COMPUTER_AND_SOFTWARE	Ordinateurs et logiciels
HEALTH_AND_BEAUTY	Santé et beauté
SERVICE_FOR_INDIVIDUAL	Services à la personne
SERVICE_FOR_BUSINESS	Services aux entreprises
SPORTS	Sports
CLOTHING_AND_ACCESSORIES	Vêtements et accessoires
TRAVEL	Voyage
HOME_AUDIO_PHOTO_VIDEO	Son, image et vidéo
TELEPHONY	Téléphonie



## 21.5. Objet Marketplace

Nom	Description	Format	Exemple
uuid	Identifiant unique de la Marketplace <i>Généré par Lyra Collect</i>	an..36	6ea5e574-e198-55d4-ba23-f9405ec4226b
href	URL d'accès à cette ressource <i>Généré par Lyra Collect</i>	cf. exemple	https://secure.lyra.com/marketplace/marketplaces/6ea5e574-e198-55d4-ba23-f9405ec4226b
created_at	Date et heure de création de la ressource <i>Généré par Lyra Collect</i>	cf. exemple	2015-01-17T09:39:54.948Z
updated_at	Date et heure de la dernière modification de la ressource <i>Généré par Lyra Collect</i>	cf. exemple	2015-01-17T09:39:54.948Z
reference	Nom de la boutique Lyra Collect créée pour la Marketplace	an..255	Ex :Nom de la boutique Marketplace
description	Nom de la boutique Lyra Collect créée pour la Marketplace	an..255	Ex :Nom de la boutique Marketplace
bic	Code banque du compte bancaire de la Marketplace	an..11	CMCIC1234
iban	Numéro de compte du compte bancaire de la Marketplace	an..34	FR1212341234123412341234
vads_key	Identifiant de la boutique Lyra Collect de la Marketplace <i>Généré par Lyra Collect</i>	an..8	12345678
vads_cert	Certificat de production Lyra Collect de la Marketplace <i>Généré par Lyra Collect</i>	an..16	123456789123456789
webhook_url	URL de notification de la Marketplace	an..1024	http://marketplace.com/page_traitement_notification.php
status	Statut de la ressource Marketplace <i>Généré par Lyra Collect</i>	an..10	ACTIVE
links	Liens vers les ressources Sellers et Orders de la Marketplace <i>Généré par Lyra Collect</i>	cf. exemple	<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>

## 21.6. Objet Order

Nom	Description	Format	Obligatoire	Exemple
uuid	Identifiant unique de la commande <i>Généré par Lyra Collect</i>	an..36	N/A	8ea5e574-e198-55d4-ba23-f9405ec4226c
href	URL d'accès à cette ressource <i>Généré par Lyra Collect</i>	cf. exemple	N/A	<a href="https://secure.lyra.com/marketplace/orders/8ea5e574-e198-55d4-ba23-f9405ec4226c">https://secure.lyra.com/marketplace/orders/8ea5e574-e198-55d4-ba23-f9405ec4226c</a>
created_at	Date et heure de création de la ressource <i>Généré par Lyra Collect</i>	cf. exemple	N/A	2015-01-17T09:39:54.948Z
updated_at	Date et heure de la dernière modification de la ressource <i>Généré par Lyra Collect</i>	cf. exemple	N/A	2015-01-17T09:39:54.948Z
marketplace	UUID de la Marketplace	an..36	X	6ea5e574-e198-55d4-ba23-f9405ec4226b
reference	Référence de la commande	an..32+"-"	X	CMD-123456
description	Description de la commande	an..255		Commande 123456 - Margaritas x2 + 4saisons x1
vads_transaction_id	Identifiant de la transaction Lyra Collect	n..6	N/A	379 <i>Remarque : Champ déprécié à cet endroit.</i>
vads_transaction_date	Date et heure de la transaction Lyra Collect au format YYYYMMDDHHMMSS	an..14	N/A	20150320094512 <i>Remarque : Champ déprécié à cet endroit.</i>
buyer	Informations sur l'acheteur	Objet Buyer	X	
alias	Identifiant du token dans la plateforme	ans..64		1144951ea7ab42989c97159b3dfc0382
shipping	Informations sur la livraison	Objet Shipping		
amount	Montant total de la commande, exprimé dans la plus petite unité de la devise.	n..12	N/A	50000 pour 500 EUR.
currency	Devise de la commande (ISO 4217)	a3	X	EUR
language	Langue d'affichage des pages de paiement (ISO 639-1)	a2		FR
capture_delay	Indique le délai en nombre de jours avant remise en banque	Entier positif compris entre 0 et 6	Non. Seulement disponible en création (POST)	3
expected_capture_date	Date de remise en banque souhaitée, dans le fuseau UTC, au format YYYYMMDDHHMMSS.	Date	Non. Seulement disponible en mise à jour (PUT)	20180716083000
status	Statut de la commande <i>Généré par Lyra Collect</i>	an..10	N/A	PENDING
webhook_result	Résultat de l'envoi de la notification à la Marketplace <i>Généré par Lyra Collect</i>	n..3	N/A	200
url_return	URL par défaut de retour de l'acheteur sur le site de la Marketplace.	an..512		<a href="http://www.sitemarketplace.com/url_de_retour.html">http://www.sitemarketplace.com/url_de_retour.html</a>
url_success	URL de retour de l'acheteur sur le site de la Marketplace après un paiement en succès.	an..200		<a href="http://www.sitemarketplace.com/url_success.html">http://www.sitemarketplace.com/url_success.html</a>

Nom	Description	Format	Obligatoire	Exemple
url_refused	URL de retour de l'acheteur sur le site de la Marketplace après un paiement refusé.	an..200		<a href="http://www.sitemarketplace.com/url_refused.html">http://www.sitemarketplace.com/url_refused.html</a>
url_cancel	URL de retour de l'acheteur sur le site de la Marketplace après un paiement annulé.	an..200		<a href="http://www.sitemarketplace.com/url_cancel.html">http://www.sitemarketplace.com/url_cancel.html</a>
url_error	URL de retour de l'acheteur sur le site de la Marketplace après un paiement en erreur.	an..200		<a href="http://www.sitemarketplace.com/url_error.html">http://www.sitemarketplace.com/url_error.html</a>
return_mode	Méthode de transmission des données utilisée lors du retour vers le site marchand.	a4		POST
links	Liens vers les ressources Items <i>Générés par Lyra Collect</i>	cf. exemple	N/A	<pre>{   "items": {     "href":       "https://secure.lyra.com/       marketplace/orders/8ea5e574-       e198-55d4-       ba23-f9405ec4226c/items"   } }</pre>
persist_url	Lien de paiement persistant	url	N/A	"https://secure.lyra.com/marketplace/t/ioeD1uRP"
expiry_date	Date d'expiration du lien persistant	cf. exemple	N/A	"2021-02-15T15:37:47.632009Z"
form_token	Token du formulaire embarqué	an		Exemple: ffa0yfV2FQuR3aak1SwsgcuZg195eyJhbW9OjEwOTAwLCJjdXJyZW5jeSI6kVVUilsIm1vUiOiJURVNUliwidmVyc2lvbil6Mywib3JkZXJJl6ljvJNDk1N2E5LWNkOGEtNGJhZC1hMGNgxMTZiMWE3ZDdmNiIsInNob3BOYW1lIjoiyYSBTTVMiLCJicmFuZFByaW9yaXR5IjpbIkJTkNPTIRBQ1QiLCJkQjlsIkUtQ0FSVEVCTEVVRSIsIk1BU1RFUkNBUkQiLCVGVQVVMVCj9fX0a702

## 21.7. Objet Order voucher

Nom	Description	Format	Obligatoire	Exemple
contract_type	Type de contrat	"CONECS" ou "CVCONNECT"	X	"CONECS"
eligible_amount	Montant éligible dans la plus petite unité monétaire de la devise	n..12	Obligatoire si contract_type = "CONECS"	1900

## 21.8. Objet Refund

Nom	Description	Format	Obligatoire
order	Identifiant de la commande à rembourser	an..36	X
reference	Référence technique du remboursement	an..32	X
status	Statut du remboursement	an..10	X
description	Description du remboursement	ans..255	
currency	Devise de la commande (ISO 4217) Seul la valeur <b>EUR</b> est autorisée	a3	X
items	Liste des parts du remboursement à effectuer, par vendeur	enum	X

Nom	Description	Format	Obligatoire
<b>items.seller</b>	Identifiant du vendeur devant assumer cette part du remboursement	an..36	X
<b>items.reference</b>	Référence technique	an..32	X
<b>items.description</b>	Description	ans..255	
<b>items.amount</b>	Montant de la part à rembourser par le vendeur	n..12	X
<b>items.item</b>	Référence de l'item sur lequel porte le remboursement au sein de l'order d'origine	an..50	

## 21.9. Objet Seller

---

Nom	Description	Format	Exemple
uuid	Identifiant unique du prestataire <i>Généré par Lyra Collect</i>	an..36	<i>1ea5e574-e198-55d4-ba23-f9405ec4226a</i>
href	URL d'accès à cette ressource <i>Généré par Lyra Collect</i>	cf. exemple	<i>https://secure.lyra.com/marketplace/sellers/1ea5e574-e198-55d4-ba23-f9405ec4226a</i>
created_at	Date et heure de création de la ressource <i>Généré par Lyra Collect</i>	cf. exemple	<i>2015-01-17T09:39:54.948Z</i>
updated_at	Date et heure de la dernière modification de la ressource <i>Généré par Lyra Collect</i>	cf. exemple	<i>2015-01-17T09:39:54.948Z</i>
marketplace	UUID de la Marketplace	an..36	<i>6ea5e574-e198-55d4-ba23-f9405ec4226b</i>
reference	Identifiant de la boutique Lyra Collect du prestataire	an..255	<i>87654321</i>
description	Nom de la boutique Lyra Collect du prestataire	an..255	<i>Boutique du prestataire</i>
bic	Code banque du compte bancaire du prestataire	an..11	<i>CMCIC1234</i>
iban	Numéro de compte du compte bancaire du prestataire	an..34	<i>FR1212341234123412341234</i>
is_marketplace_seller	ce champ permet d'identifier le seller de type market place. Dans la commande les items de type commission (is_commission = true) doivent être rattachés à ce seller. Ce seller est créé automatiquement lors de la création de la marketplace.	booléen	<i>true   false</i>
status	Statut de la ressource Marketplace <i>Généré par Lyra Collect</i>	an.10	<i>ACTIVE</i>
links	Liens vers les ressources Items du prestataire <i>Générés par Lyra Collect</i>	cf. exemple	<pre>{   "items": {     "href":     "https://secure.lyra.com/marketplace/sellers/1ea5e574-e198-55d4-ba23-f9405ec4226a/items"   }, }</pre>

## 21.10. Objet Shipping

---

Nom	Description	Format	Obligatoire	Exemple
delivery_company_na	nom du transporteur	ans..128		
shipping_speed	mode de livraison	enum		<i>STANDARD   EXPRESS   PRIORITY</i>
shipping_method	méthode de livraison	enum	X	<i>RECLAIM_IN_SHOP   RELAY_POINT   RECLAIM_IN_STATION   PACKAGE_DELIVERY_COMPANY   ETICKET</i>
shipping_delay	rapidité liée au mode de livraison lorsque shipping_speed = PRIORITY	enum		<i>INFERIOR_EQUALS   SUPERIOR   IMMEDIATE   ALWAYS</i>
type	type de destinataire	enum		<i>PRIVATE   COMPANY</i>
legal_name	nom de la société	an..63		<i>Dupond &amp; Cie</i>
name	nom de la personne livrée			
first_name	prénom de l'acheteur	an..63		<i>Stéphanie</i>
last_name	nom de l'acheteur	an..63		<i>DUPOND</i>
phone_number	numéro de téléphone de l'acheteur	an..32		<i>0612324565</i>
address	adresse postale de l'acheteur	objet Address		

## 21.11. Objet Token

Nom	Description	Format	Obligatoire	Exemple
uuid	Identifiant unique de la commande <i>Généré par Lyra Collect</i>	an..36	N/A	<i>c730def9-e171-4f4f-92de-b7f09ed74b8d</i>
created_at	Date et heure de création de la ressource <i>Généré par Lyra Collect</i>	cf. exemple	N/A	<i>2018-11-27T13:53:48.461191Z</i>
updated_at	Date et heure de la dernière modification de la ressource <i>Généré par Lyra Collect</i>	cf. exemple	N/A	<i>2018-11-27T13:53:48.461191Z</i>
marketplace	UUID de la Marketplace	an..36	N/A	<i>2434c0a2-9d46-4e96-9553-1536c898625b</i>
buyer	Informations sur l'acheteur	Objet Buyer	X	
language	Langue d'affichage des pages de paiement (ISO 639-1)	a2		<i>FR</i>
url_return	URL de retour de l'acheteur sur le site de la Marketplace	an..512		<i>http://www.my-website.com</i>
status	Statut de la demande <i>Généré par Lyra Collect</i>	an..10	N/A	<i>CREATED</i>
alias	Identifiant du token dans la plateforme	ans..64		<i>1144951ea7ab42989c97159b3dfc0382</i>
alias_to_update	Identifiant du token à modifier dans la plateforme	ans..64		<i>1144951ea7ab42989c97159b3dfc0382</i>
payment_url	URL à donner au client pour la création du token	ans..255		

## 21.12. Objet Transaction

Nom	Description	Format	Obligatoire	Exemple
<b>trans_uuid</b>	Identifiant de la transaction	ans..32	X	03934e6df4ec4f8ea0fe491e95fbc619
<b>order</b>	Identifiant de la commande associée		X	33ec0e5a-f25c-4bb1-8115-e54c70e30f16
<b>sequence_number</b>	Indice de la transaction, notamment utile dans le cas des paiement en N fois.	n..12	X	1 pour la première transaction, 2 pour la seconde, etc.
<b>paid_at</b>	Date du paiement.	Date	Non	2020-06-15T13:16:22Z
<b>auto_code</b>	Code retour autorisation.	a..2	Non	Voir <i>ci-dessous</i> la liste des valeurs.
<b>payment_scheme</b>	Description du moyen de paiement employé.	an..16	N/A	CB
<b>amount</b>	Montant de la transaction, exprimé dans la plus petite unité de la devise.	n..12	X	10450 pour 104,50 EUR
<b>initial_amount</b>	Montant de la transaction enregistré lors de la demande d'autorisation, exprimé dans la plus petite unité de la devise.	n..12	Non	10450 pour 104,50 EUR
<b>expected_capture_date</b>	Date de remise en banque souhaitée, dans le fuseau UTC, au format YYYYMMDDHHMMSS.	Date	X	20200615151622
<b>status</b>	Statut de la transaction. <i>Généré par Lyra Collect</i>	an..10	Non	TO_VALIDATE
<b>links</b>	Liens vers les ressources transactions. <i>Généré par Lyra Collect</i>	cf. exemple	Non	"links": { "transfers": { "href": "https://secure.lyra.com/marketplace/transactions/03a0f95fbc619/transfers" } }
<b>created_at</b>	Date et heure de création de la ressource.	cf. exemple	Non	2020-06-15T13:16:49.223042Z
<b>updated_at</b>	Date et heure de la dernière modification de la ressource.	cf. exemple	Non	



## Liste des codes retour autorisation (auto\_code) :

Codes retournés par l'acquéreur **Amex Global**:

Code	Description
000	Approuvée
001	Approuvée avec pièce d'identité
002	Autorisation partielle (Cartes prépayées seulement)
100	Refusée
101	Carte expirée / Date d'expiration invalide
106	Nombre d'essais permis de saisie du NIP dépassé
107	Veuillez appeler l'émetteur
109	Marchand invalide
110	Montant invalide
111	Compte invalide / MICR invalide
115	Fonction demandée non prise en charge
117	NIP invalide
119	Titulaire non inscrit / non permis
122	Code de sécurité de la carte invalide (alias NIC/C4C)
125	Date d'entrée en vigueur invalide
130	Refusée
181	Erreur de format
183	Code de devise invalide
187	Refusée - Nouvelle carte émise
189	Refusée - Compte annulé
200	Refusée - Reprendre Carte
900	Acceptée - Synchronisation ATC
909	Dysfonctionnement du système (erreur cryptographique)
912	Émetteur non disponible

Codes retournés sur le réseau **CB** et **Paylib** :

Valeur	Description	Valeur	Description
00	Transaction approuvée ou traitée avec succès	43	Carte volée
02	Contactez l'émetteur de carte	51	Provision insuffisante ou crédit dépassé
03	Accepteur invalide	54	Date de validité de la carte dépassée
04	Conserver la carte	55	Code confidentiel erroné
05	Ne pas honorer	56	Carte absente du fichier
07	Conserver la carte, conditions spéciales	57	Transaction non permise à ce porteur
08	Approuver après identification	58	Transaction non permise à ce porteur
12	Transaction invalide	59	Suspicion de fraude
13	Montant invalide	60	L'accepteur de carte doit contacter l'acquéreur
14	Numéro de porteur invalide	61	Montant de retrait hors limite
15	Émetteur de carte inconnu	63	Règles de sécurité non respectées
17	Annulation acheteur	68	Réponse non parvenue ou reçue trop tard
19	Répéter la transaction ultérieurement	75	Nombre d'essais code confidentiel dépassé
20	Réponse erronée (erreur dans le domaine serveur)	76	Porteur déjà en opposition, ancien enregistrement conservé
24	Mise à jour de fichier non supportée	80	Le paiement sans contact n'est pas admis par l'émetteur
25	Impossible de localiser l'enregistrement dans le fichier	81	Le paiement non sécurisé n'est pas admis par l'émetteur
26	Enregistrement dupliqué, ancien enregistrement remplacé	82	Révocation paiement récurrent pour la carte chez le commerçant ou pour le MCC et la carte
27	Erreur en « edit » sur champ de liste à jour fichier	83	Révocation tous paiements récurrents pour la carte

Valeur	Description	Valeur	Description
28	Accès interdit au fichier	90	Arrêt momentané du système
29	Mise à jour impossible	91	Émetteur de cartes inaccessible
30	Erreur de format	94	Transaction dupliquée
31	Identifiant de l'organisme acquéreur inconnu	96	Mauvais fonctionnement du système
33	Date de validité de la carte dépassée	97	Échéance de la temporisation de surveillance globale
34	Suspicion de fraude	98	Serveur indisponible routage réseau demandé à nouveau
38	Date de validité de la carte dépassée	99	Incident domaine initiateur
41	Carte perdue		

#### Codes retournés sur le réseau **GICC**:

Code	Description
0	Approved or completed successfully
2	Call Voice-authorization number; Initialization Data
3	Invalid merchant number
4	Retain card
5	Authorization declined
10	Partial approval
12	Invalid transaction
13	Invalid amount
14	invalid card
21	No action taken
30	Format Error
33	Card expired
34	Suspicion of Manipulation
40	Requested function not supported
43	Stolen Card, pick up
55	Incorrect personal identification number
56	Card not in authorizer's database
58	Terminal ID unknown
62	Restricted Card
78	Stop payment order
79	Revocation of authorization order
80	Amount no longer available
81	Message-flow error
91	Card issuer temporarily not reachable
92	The card type is not processed by the authorization center
96	Processing temporarily not possible
97	Security breach - MAC check indicates error condition
98	Date and time not plausible
99	Error in PAC encryption detected

#### Codes retournés sur le réseau **PayPal**

Code	Description
0	Transaction accepted
10001	Internal error
10002	Restricted Account
10009	Transaction refused for one of the following reasons: <ul style="list-style-type: none"> <li>The partial refund amount must be less than or equal to the original transaction amount.</li> <li>The partial refund must be the same currency as the original transaction.</li> <li>This transaction has already been fully refunded.</li> </ul>

Code	Description
	<ul style="list-style-type: none"> <li>You are over the time limit to perform a refund on this transaction (60 days).</li> </ul>
10422	Customer must choose new funding sources. The customer must return to PayPal to select new funding sources.
10486	This transaction couldn't be completed. Please redirect your customer to PayPal.
13113	The Buyer cannot pay with PayPal for this transaction. Inform the buyer that PayPal declined the transaction and to contact <a href="#">PayPal Customer Service</a> .

#### Codes retournés par l'acquéreur **Edenred Belgique**

Code	Description
<b>0</b>	Paiement accepté
<b>1</b>	Approuvée partiellement. L'acheteur a été invité à régler le restant dû avec un autre moyen de paiement.
<b>5</b>	Paiement refusé
<b>102</b>	Numéro de carte invalide
<b>103</b>	Détaillant non autorisé
<b>104</b>	Carte bloquée
<b>105</b>	Fonds insuffisants
<b>106</b>	Carte expirée
<b>107</b>	Pin incorrect
<b>108</b>	Erreur non référencée
<b>114</b>	Carte non activée
<b>115</b>	Solde dépassé
<b>116</b>	Devise non valide
<b>117</b>	Plafond journalier dépassé
<b>118</b>	Plafond hebdomadaire dépassé
<b>119</b>	Plafond mensuel dépassé
<b>120</b>	Plafond annuel dépassé
<b>121</b>	Nombre d'essais du code confidentiel dépassé
<b>122</b>	Montant trop faible
<b>123</b>	Montant trop élevé
<b>124</b>	Compte bloqué
<b>125</b>	Limite du montant cumulé dépassé
<b>126</b>	Limite du nombre d'achats dépassé

## 22. OBTENIR DE L'AIDE

---

Vous cherchez de l'aide? Consultez notre FAQ sur notre site

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

Pour toute question technique ou demande d'assistance, contactez [le support technique](#).

Pour faciliter le traitement de vos demandes, il vous sera demandé de communiquer votre code client (Exemple : **CLXXXXX**, **MKXXXXXX** ou **AGXXXXXX**).

Cette information est disponible dans le Back Office Marchand (en haut du menu).